UNCLASSIFIED

AD NUMBER
AD872814
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; JUL 1970. Other requests shall be referred to Army Aviation Materiel Labs., Fort Eustis, VA.**
AUTHORITY
USAAVLABS ltr 23 Jun 1971

THIS PAGE IS UNCLASSIFIED

AD

AD872814

AD No. ____

DDC FILE COPY

USAAVLABS TECHNICAL REPORT 70-15

STRESS DISTRIBUTION AND MATERIAL CHARACTERISTICS OF COMPOSITE MATERIALS UNDER OBLIQUE LOADINGS

By

Juan Haener

Ming-yuan Feng

July 1970

U. S. ARMY AVIATION MATERIEL LABORATORIES
FORT EUSTIS, VIRGINIA

CONTRACT DAAJ02-69-C-0029
WHITTAKER CORPORATION
RESEARCH AND DEVELOPMENT DIVISION
SAN DIEGO, CALIFORNIA

183

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission, to manufacture, use, or sell any patented invention that may in any way be related thereto.

Disposition Instructions

Destroy this report when no longer needed. Do not return it to the originator.

DEPARTMENT OF THE ARMY
HEADQUARTERS US ARMY AVIATION MATERIEL LABORATORIES
FORT EUSTIS, VIRGINIA 23604

This program was performed under Contract DAAJ02-69-C-0029 with Whittaker Corporation, San Diego, California.

The data contained in this report are the result of research concerned with the load transfer through a composite with unidirectional fibers embedded in a matrix under consideration of oblique loading.

The report has been reviewed by this command and is considered to be technically sound; however, experimental verification has not been completed, and the information should be used accordingly. It is published for the exchange of information and the stimulation of future research.

Task 1F162204A17002
Contract DAAJ02-69-C-0029
USAAVLABS Technical Report 70-15
July 1970

STRESS DISTRIBUTION AND MATERIAL CHARACTERISTICS OF COMPOSITE MATERIALS UNDER OBL.QUE LOADINGS

Final Report

By

Juan Haener

and

Ming-yuan Feng

Prepared by

Whittaker Corporation
Research & Development Division
San Diego, California

for

U. S. ARMY AVIATION MATERIEL LABORATORIES
FORT EUSTIS, VIRGINIA

ABSTRACT

The critical loadings at failure of a unidirectional fibrous composite under any oblique loading are the most important results obtained in this work. For this purpose the von Mises energy criterion was applied to the components of the composite. Debonding failures at the interfaces were also considered.

Other results of this research include composite elastic engineering constants in any angular direction as a function of fiber density and component properties.

Diagrams are presented which exhibit the critical loading and the elastic coefficients of a series of composites as a function of the loading directions, component material constants, and geometry.

The fundamentals to this work are based on the micromechanical stress fields in the fibers and in the matrix.

FOREWORD

This final report was prepared by Whittaker Corporation, Research and Development Division, under U. S. Army Contract DAAJ02-69-C-0029 (Task 1F162204A17002) for the U. S. Army Aviation Materiel Laboratories, Fort Eustis, Virginia. Mr. A. Gustafson, Jr., was the Army Project Officer for this program.

This report covers the work accomplished during the period from 31 October 1968 through 31 October 1969.

Acknowledgement is given to Dr. Gerhard Nowak for his assistance in the development of boundary conditions of the composite under transverse shear, to Dr. Harold Evensen for his analysis on the symmetry relations of perforated plates, and to Mr. Ted Neff for writing the computer program for the numerical analysis.

BLANK PAGE

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF SYMBOLS

a	Radius of fiber in inches
$\{a\}$	Column vector of the coefficients relating displacement components to local coordinates of the nodal points
$[A]$	Coordinates matrix
$A_{x,i}$	Denoting the area of the plane of triangular prism i along the boundary x = c in the basic representative element
$A_{y,j}$	Denoting the area of the plane of triangular prism j along the boundary y = b/2 in the basic representative element
$A_{z,k}$	Denoting the area of the plane of triangular prism k along the boundary z = h in the basic representative element
$A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2, C_3$	Constants
b	Half the distance between the axes of two neighboring fibers in inches; also the dimension of a basic representative element in y-direction
c	Half of the dimension in inches of a basic representative element in x-direction
$[d]$	Direction cosine column vector of the direction normal d at the interface
$[D]$	Coefficients matrix relating strain components of coefficients in the expression of displacements in psi
E	Major Young's modulus of the composite in the direction of loading p in psi
E_{11}, E_r	Young's modulus of the composite in the direction perpendicular to the fiber axis in psi
E_{33}, E_z	Young's modulus of the composite in the direction of fiber axis in psi

LIST OF SYMBOLS (Continued)

E_f	Young's modulus of fiber in psi
E_m	Young's modulus of matrix in psi
G	Shear modulus of the composite in the plane parallel or perpendicular to the direction of the loading p in psi
G_{31}	Shear modulus of the composite in relation to plane normal to plane 3 in direction 1 or z = x axes in psi
G_f	Shear modulus of fiber in psi
G_m	Shear modulus of matrix in psi
h	Height of a basic representative element in z-direction in inches
i,j,k	Representing natural sequential integers initialized to 1
I,J,K	Denoting the total numbers of triangular elements along the boundaries x = c, y = b/2 and z = h respectively (in the present analysis I = 6, J = 11, K = 158)
m	$=\cos\theta$
n	$=\sin\theta$
[N]	Macrostress column vector due to a generalized external loading(s)
$N_x, N_y, N_z, T_{yz}, T_{zx}, T_{xy}$	Macrostress components due to a generalized external loading(s)
$\bar{N}_x, \bar{N}_y, \bar{N}_z, \bar{T}_{yz}, \bar{T}_{zx}, \bar{T}_{xy}$	Macrostress components (fundamental loadings) actually used as the inputs of a basic representative element for six fundamental cases
P	External oblique loading on x'-z' (or 4'-3') plane in psi

$(p_{cr})_c$	Critical loading calculated by von Mises'-Hencky Distortion Energy Theory applied at the triangular prisms of the constituents of the composite in psi
$(p_{cr})_{is}$	Critical loading based on the interfacial shear bonding strength between fiber and matrix in psi
$(p_{cr})_{in}$	Critical loading based on the normal bonding strength at the interface between fiber and matrix in psi
[S]	6 x 6 matrix of stress components of six fundamental loadings
$S_{11}, S_{21}, S_{31}, S_{41}, S_{51}, S_{61}$	Stress components produced in the composite due to unity transverse loading $N_x (S_{41}=S_{51}=0)$ (Revised Fundamental Case N_y)
$S_{12}, S_{22}, S_{32}, S_{42}, S_{52}, S_{62}$	Stress components produced in the composite due to unity transverse loading $N_y (S_{42}=S_{32}=0)$ (Revised Fundamental Case N_z)
$S_{13}, S_{23}, S_{33}, S_{43}, S_{53}, S_{63}$	Stress components produced in the composite due to axial loading $N_z (S_{43}=S_{53}=0)$ (Revised Fundamental Case N_z)
$S_{14}, S_{24}, S_{34}, S_{44}, S_{54}, S_{64}$	Stress components produced in the composite due to unity longitudinal shear T_{yz} in Revised Fundamental Case $T_{yz} (S_{14}=S_{24}=S_{34}=S_{64}=0)$
$S_{15}, S_{25}, S_{35}, S_{45}, S_{55}, S_{65}$	Stress components produced in the composite due to unity longitudinal shear T_{zx} in Revised Fundamental Case $T_{zx} (S_{15}=S_{25}=S_{35}=S_{65}=0)$
$S_{16}, S_{26}, S_{36}, S_{46}, S_{56}, S_{66}$	Stress components produced in the composite due to unity transverse shear T_{xy} in Revised Fundamental Case $T_{xy} (S_{46}=S_{56}=0)$

LIST OF SYMBOLS (Continued)

$\bar{S}_{11},\bar{S}_{21},\bar{S}_{31}$, $\bar{S}_{41},\bar{S}_{51},\bar{S}_{61}$	Stress components produced in the composite due to unity transverse normal loading $\bar{N}_x$ in Fundamental Case $N_x(\bar{S}_{41}=\bar{S}_{51}=0)$
$\bar{S}_{12},\bar{S}_{22},\bar{S}_{32}$, $\bar{S}_{42},\bar{S}_{52},\bar{S}_{62}$	Stress components produced in the composite due to unity transverse normal loading $\bar{N}_y$ in Fundamental Case $N_y(\bar{S}_{42}=\bar{S}_{52}=0)$
$\bar{S}_{13},\bar{S}_{23},\bar{S}_{33}$, $\bar{S}_{43},\bar{S}_{53},\bar{S}_{63}$	Stress components produced in the composite due to unity axial loading $\bar{N}_z$ in Fundamental Case $N_z(\bar{S}_{43}=\bar{S}_{53}=0)$
$\bar{S}_{14},\bar{S}_{24},\bar{S}_{34}$, $\bar{S}_{44},\bar{S}_{54},\bar{S}_{64}$	Stress components produced in the composite due to unity longitudinal shear $\bar{T}_{yz}$ in Fundamental Case $T_{yz}(\bar{S}_{14}=\bar{S}_{24}=\bar{S}_{34}=\bar{S}_{64}=0)$
$\bar{S}_{15},\bar{S}_{25},\bar{S}_{35}$, $\bar{S}_{45},\bar{S}_{55},\bar{S}_{65}$	Stress components produced in the composite due to unity longitudinal shear $\bar{T}_{xy}$ in Fundamental Case $T_{yz}(\bar{S}_{15}=\bar{S}_{25}=\bar{S}_{35}=\bar{S}_{65}=0)$
$\bar{S}_{16},\bar{S}_{26},\bar{S}_{36}$, $\bar{S}_{46},\bar{S}_{56},\bar{S}_{66}$	Stress components produced in the composite due to unity transverse shear $\bar{T}_{xy}$ in Fundamental Case $T_{xy}(\bar{S}_{16}=\bar{S}_{26}=0)$
$[T]$	Matrix of stress components
$[\ ]^T$	Transposed matrix
$[T_d]$	Stress column vector associated with the direction normal at the interface
$\{u\}$	Displacement column vector at nodal points
U_d	Denoting distortion energy at a point in the matrix due to actual combined loadings
U_{df}	Denoting distortion energy at failure due to a simple tension test
u,v,w	Displacement components in x,y,z-directions respectively

LIST OF SYMBOLS (Continued)

V_f	Percentage of volumetric content of fiber(s) in a composite
V	Volume of the triangular prism
x,y,z	Cartesian coordinates in right-handed coordinate system, with z (or 3) axis parallel to fiber axis
x',y',z' or $1',2',3'$	Cartesian coordinates in right-handed coordinate system, with z'(or 3') axis parallel to the direction of loading p
α	Arc tangent of the ratio between τ_{yz} and τ_{xz}
β	Ratio of Poisson's ratios between fiber and matrix
γ	Ratio of Young's modulus between fiber and matrix
$\{\varepsilon\}$	Strain components
$\{\varepsilon_1\}$	Strain column vector defined by equation (5)
$\{\varepsilon_2\}$	Strain column vector defined by equation (6)
η	Coupling factor
θ	Angle between directions of fiber axis and loading p in x = z (or 4-3, x'-z' or 1'-3') plane
ν	Major Poisson's ratio of a unidirectional fiber-reinforced composite due to an oblique loading p
ν_{31}	Poisson's ratio due to loading in the direction of fiber axis
ν_{13}	Poisson's ratio due to loading in the direction perpendicular to fiber axis
ν_f	Poisson's ratio of fiber
ν_m	Poisson's ratio of matrix
$\{\sigma\}$	Stress column vector

LIST OF SYMBOLS (Continued)

$\sigma_x, \sigma_y, \sigma_z, \tau_{yz}, \tau_{zx}, \tau_{xy}$	Microstress components in a composite due to a generalized loading p in psi
$\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6$	Microstress components in contracted notation, corresponding to $\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \tau_{yz}, \tau_{zx}, \tau_{xy}$
$\sigma_x', \sigma_y', \sigma_z', \tau_{xy}', \tau_{xz}', \tau_{yz}'$	Externally applied stress components (in the present analysis, $\sigma_z' = p$, others equal to 0)
$\sigma_{x,i}$	Denoting the normal stress in x-direction of triangular prism i along the boundary x = c, due to the boundary conditions used in the computation program in fundamental case N_x
$\sigma_{y,j}$	Denoting the normal stress in y-direction of triangular prism j along the boundary y = b/2 due to the boundary condition used in the computation program in fundamental case N_y
$\sigma_{z,k}$	Denoting the normal stress in z-direction of triangular prism k along the boundary z = h, due to the boundary conditions used in the computation program in fundamental case N_z
σ_f	Failure strength of the material due to a simple tension or compression test in psi
σ_{fc}	Yield strength in compression of the material due to a simple compression test in psi
σ_{ft}	Yield strength in tension of the material due to a simple tension test in psi
σ_n	Normal stress at the interface in psi
σ_{nf}	Normal failure debonding strength at the interface between fiber and matrix
τ_t	Tangential stress at the interface in psi
τ_{tf}	Interfacial tangential fibers debonding strength between fiber and matrix in psi

LIST OF SYMBOLS (Continued)

$\tau_{yz,j}$	Denoting the shear stress of triangular prism j along the boundary y=b/2, due to the boundary conditions used in the computation program in fundamental case T_{yz}
$\tau_{zx,i}$	Denoting the shear stress of triangular prism i along the boundary x=c, due to the boundary conditions used in the computation program in fundamental case T_{zx}
$\tau_{xy,i}$	Denoting the shear stress of triangular prism i along the boundary x=c, due to the boundary conditions used in the computation program in fundamental case T_{xy}
Ω	Area of the triangle of each triangular prism

BLANK PAGE

INTRODUCTION

In ideal cases only, the loading of a composite material is in the direction of the reinforcements. Normally, oblique loading is felt by the composite of which a structural component is made. Examples are materials consisting of laminates in which each laminate has a definite fiber orientation. Other examples where oblique loading is involved are the different kinds of mechanical and other joints.

Therefore, the problem to be investigated is a unidirectional fiber reinforced composite under generalized oblique loading.

To obtain the effect of an oblique loading in any direction to the fiber axis, six fundamental loadings have to be combined. Based on micro-mechanical considerations and by linear superposition (References 1 and 2) of stresses due to the six fundamental loadings, the combined micro-stresses are then computed. References 3, 4, and 5 give the analyses of the following six fundamental loading cases: axial loading, transverse normal (two cases), longitudinal shear, and transverse shear (two cases)*. Then, having the combined stresses which are acting on each particle of the reinforcement and the matrix, a failure criterion can be adopted. Here the von Mises distortion energy theory was used to calculate the critical loading in each element. For the points at the interfaces between fibers and matrix, debonding failures have to be evaluated in normal as well as tangential directions of the curved surfaces. It is self-evident that only the smallest critical loading of a composite has to be determined by the above outlined procedure. Which of the particles under stress will be first subjected to failure depends on geometry and material combinations and will be automatically considered. Failure may occur anywhere at the interface, in the fiber or in the resin.

The elastic engineering constants computed for each fundamental loading case will then be used to obtain, through transformation laws for fourth-order tensors, the composite elastic constants in any direction of the composite.

---

* Later it will be explained why two cases are necessary in the transverse direction.

TECHNICAL DISCUSSION

In the present problem the internal microstresses produced by external oblique loading can be envisioned as linear superimposed stresses produced by six independent fundamental loadings. It was assumed that the fiber packing is hexagonal since a model of such an array provides, as explained in Reference 6, better agreement with experiments than any other model. Therefore, the oblique loading can be produced by the following fundamental loadings:

Transverse normal loadings	N_x, N_y
Axial loading	N_z
Longitudinal shear loadings	N_{yz}, T_{zx}
Transverse shear loading	T_{xy}

Figure 1 shows these fundamental loadings acting on the surface of the representative element.

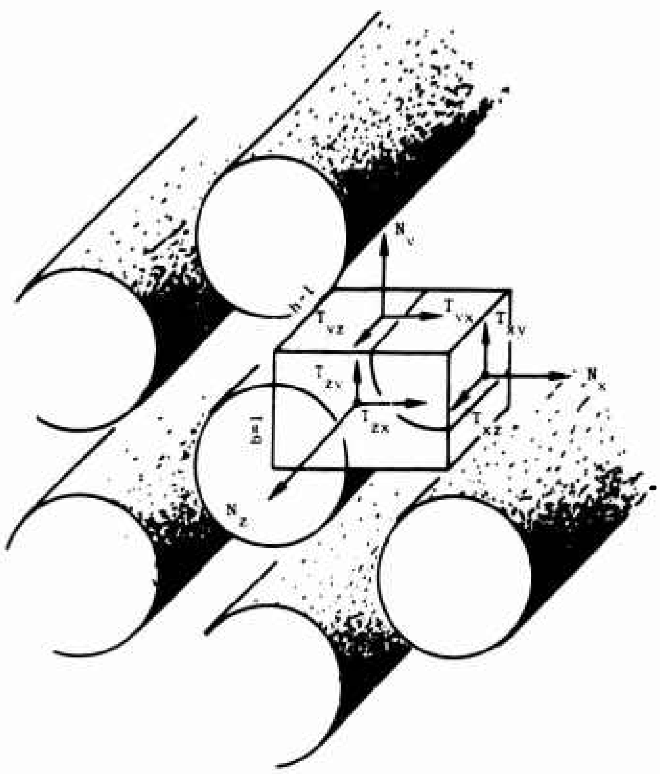


Figure 1. Basic Representative Element and the Loadings on the Surfaces of the Element.

The stress tensor is assumed to be symmetric, $T_{ij} = T_{ji}$, so that only six of the nine loads shown in Figure 1 have to be considered. Since external loadings are usually inclined at an angle with the fiber axis, the loadings as shown are the outputs of external loadings actually applied to the surface of the composite. For this purpose the transformation law for rank two tensors (References 7 and 8) has been applied. At this state of manipulation, the transformation law is valid.

These so-obtained loadings are the load boundary conditions for the combined internal microstresses which they produce. In other words, these loadings are the inputs of the analysis of the fundamental cases for which the microstresses will be deduced.

The microstresses are then added linearly in each particle of the reinforcement and matrix and introduced into the von Mises criterion so that the critical load in the constituents due to combined loading of a component is obtained.

Instead of beginning with the transformation of the oblique load into fundamental loads, we present first the latter ones. The finite element method itself will not be repeated here since this technique was already explained in detail in Reference 3.

TRANSVERSE NORMAL LOADINGS, N_x AND N_y

The fundamental load case N_x was solved by finite element methods as a three-dimensional as well as a two-dimensional problem in Reference 3. It was found that accuracy of a two-dimensional analysis is sufficient for continuous fiber composites. In the present work, additional two-dimensional plane-strain solutions have been obtained separately for the N_x and N_y loadings.

The boundary conditions used in the computation program are for transverse normal loading N_x:

1) Displacements in x-direction along the boundary planes perpendicular to x-axis are 1 and -1 respectively; i.e., $u = 1$ at $x = \frac{1}{2}\sqrt{3}$ and $u = -1$ at $x = \frac{1}{2}\sqrt{3}$.

2) Displacements in y-direction along the boundary planes perpendicular to y-axis are zeroes; i.e., $v = 0$ at $y = \pm\frac{1}{2}$.

3) Transverse shear stresses are zeroes along all boundaries.

For transverse normal loading case N_y:

1) Displacements in x-direction along the boundary planes perpendicular to x-axis are zeroes; i.e., $u = 0$ at $x = \pm\frac{1}{2}\sqrt{3}$.

2) Displacements in y-direction along the boundary planes perpendicular to y-axis are 1 and -1 respectively; i.e., $u = 1$ at $y = \frac{1}{2}$ and $v = -1$ at $y = -\frac{1}{2}$.

3) Transverse shear stresses are zeroes along all boundaries.

In order to convert the stresses caused by the above boundary conditions into the stresses corresponding to unity loadings in each case ($\overline{N}_x=\overline{N}_y=1$), the normalized factors $1/\sum_i^I \overline{\sigma}_{xi}A_{xi}$ and $\sqrt{3}/\sum_j^J \overline{\sigma}_{yj}A_{yj}$ were used in fundamental cases N_x and N_y respectively. The conversions for these and the subsequent fundamental cases are deemed to be necessary for the convenience of linear superposition.

The resulting stresses produced for case N_x are $\overline{S}_{11},\overline{S}_{21},\overline{S}_{31},\overline{S}_{41},\overline{S}_{51}$, and $\overline{S}_{61}$ ($\overline{S}_{41}=\overline{S}_{51}=0$), and for case N_y, $\overline{S}_{12},\overline{S}_{22},\overline{S}_{32},\overline{S}_{42},\overline{S}_{52}$, and $\overline{S}_{62}$ ($\overline{S}_{42}=\overline{S}_{52}=0$).*

* All stress components indicated here and hereafter are in their orders corresponding to final microstresses: $\sigma_x,\sigma_y,\sigma_z,\tau_{yz},\tau_{zx}$, and τ_{xy}.

AXIAL LOADING, N_z

The problem of an axially loaded composite was solved in Reference 4 by an analytical method with the use of different coordinate axes of reference. Therefore, in this analysis, it must be solved numerically by applying the same type of finite elements used in the previous cases and using the coordinate axis of reference as shown in Figure 1.

The present problem consists of two parts: (1) a two-dimensional plane-strain problem which is the same as the previous cases except, of course, for boundary conditions; and (2) a two-dimensional problem which produces a unit strain in the direction of fiber axis and is slightly different from the so-called generalized-plane-strain problem in that it does not produce stresses in x- and y-directions.

The equations used for calculation are presented as follows (most of the notations are the same as those used in Reference 3):

The displacements at the nodes are expressed as

$$\{u\} = [A]\{a\} \tag{1}$$

or

$$\{a\} = [A]^{-1}\{u\} \tag{2}$$

The strain components are

$$\{\varepsilon\} = [D]\{a\} \tag{3}$$

However, the strain column vector is a combination of strain components for a two-dimensional plane-strain solution and of strain components producing unit strain in fiber axis direction; i.e.,

$$\{\varepsilon\} = \{\varepsilon_1\} + \{\varepsilon_2\} \tag{4}$$

where

$$\{\varepsilon_1\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{5}$$

and

$$\{\varepsilon_2\} = \begin{Bmatrix} -\nu \\ -\nu \\ 0 \end{Bmatrix} \tag{6}$$

The stresses produced are then

$$\{\sigma\} = [C]\{\varepsilon_1\} = [C](\{\varepsilon\} - \{\varepsilon_2\}) \tag{7}$$

or

$$\{\sigma\} = [C]([D][A]^{-1}\{u\} - \{\varepsilon_2\}) \tag{8}$$

The strain energy is

$$W = \iiint \{\varepsilon\}^T \{\sigma\} t \, dx \, dy \, d\varepsilon$$

Since $\{\varepsilon\}^T\{\sigma\}$ represents energy density per unit strain, integration over strain ε must be carried out so that the strain energy is

$$W = \tfrac{1}{2} \iint (\{u\}^T ([A]^{-1})^T [D]^T [C][D][A]^{-1}\{u\}$$

$$-2\{u\}^T([A]^{-1})[D]^T[C]\{\varepsilon_2\}) t \, dxdy \qquad (9)$$

Then the force components at nodal points are

$$\{P_1\} = [K]\{u\} - \{P_2\} \qquad (10)$$

or

$$\{P_1\} + \{P_2\} = [K]\{u\} \qquad (11)$$

where

$$[K] = ([A]^{-1})^T [D]^T [C][D][A]^{-1} V \qquad (12)$$

and

$$\{P_2\} = \{u\}^T([A]^{-1})[D]^T[C]\{\varepsilon_2\} V \qquad (13)$$

The rest of the equations will be as in Reference 5 and will not be repeated here.

The boundary conditions used in the computation program were:

1) All displacement components are zeroes along the boundaries; i.e., $u = 0$ at $x = \pm\frac{1}{2}\sqrt{3}$ and $v = 0$ at $y = \pm\frac{1}{2}$.

2) Transverse shear stresses along the boundaries are zeroes.

The normalized factor used in the present case was $\sqrt{3}/(\sum_{k}^{K} \bar{\sigma}_{z,k} A_{z,k})$.

The resulting stresses produced are $\bar{S}_{13}, \bar{S}_{23}, \bar{S}_{33}, \bar{S}_{43}, \bar{S}_{53}$, and $\bar{S}_{63} (\bar{S}_{43} = \bar{S}_{53} = 0)$.

LONGITUDINAL SHEAR LOADINGS, T_{yz} and T_{zx}

The boundary value problem of a longitudinally shear-loaded composite is quite different from the previous cases and the subsequent case, although they are all two-dimensional. For the latter cases the governing differential equation for stress function is biharmonic, while for the former it is harmonic (as a torsional problem).

The displacement components at the nodal points are expressed as

$$\{w\} = [A]\{a\} \tag{14}$$

In matrix form, the displacements in the z-direction are

$$\begin{Bmatrix} w_1 \\ w_2 \\ w_3 \end{Bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & \xi_2 & \eta_2 \\ 1 & \xi_3 & \eta_3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{15}$$

$$\{a\} = [A]^{-1}\{w\} \tag{16}$$

where

$$[A]^{-1} = \begin{bmatrix} \dfrac{\xi_2\eta_3 - \xi_3\eta_2}{\Delta} & 0 & 0 \\ \dfrac{-\eta_3 - \eta_2}{\Delta} & \dfrac{\eta_3}{\Delta} & \dfrac{-\eta_2}{\Delta} \\ \dfrac{\xi_3 - \xi_2}{\Delta} & \dfrac{-\xi_3}{\Delta} & \dfrac{\xi_2}{\Delta} \end{bmatrix} \tag{17}$$

and $\Delta = \xi_2\eta_3 - \xi_3\eta_2$ (18)

The strain components are

$$\{\gamma\} = [D]\{a\} = [D][A]^{-1}\{w\} \tag{19}$$

or

$$\begin{Bmatrix} \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} \tag{20}$$

and the stress components are

$$\{\tau\} = [C]\{\gamma\} \tag{21}$$

or

$$\{\tau\} = [C][D][A]^{-1}\{w\} \tag{22}$$

Therefore, the strain energy is given by

$$W = \tfrac{1}{2}\iint\{\gamma\}^T\{\tau\}dxdy$$

$$= \tfrac{1}{2}\iint\{w\}^T([A]^{-1})^T[D]^T[C][D][A]^{-1}\{w\}dxdy \tag{23}$$

By Castigliano's second principle, we have

$$\{P\} = [K]\{w\} \tag{24}$$

where

$$[K] = ([A]^{-1})^T[D]^T[C][D][A]^{-1}\Omega \tag{25}$$

The rest of the equations will be similar to those in Reference 3 and will not be repeated here.

In the computation program, the boundary conditions used are:
Fundamental Longitudinal Shear Loading Case T_{yz}:

1) Longitudinal displacement of the boundary plane perpendicular to the y-axis in the positive direction is unity; i.e., $w = 1$ at $y = \frac{1}{2}$.

2) Longitudinal displacement of the boundary plane perpendicular to the y-axis in the negative direction is negative unity; i.e., $w = -1$ at $y = -\frac{1}{2}$.

Fundamental Longitudinal Shear Loading Case T_{zx}:

1) Longitudinal displacement of the boundary plane perpendicular to the x-axis in the positive direction is unity; i.e., $w = 1$ at $x = \frac{1}{2}\sqrt{3}$.

2) Longitudinal displacement of the boundary plane perpendicular to the x-axis in the negative direction is negative unity; i.e., $w = -1$ at $x = -\frac{1}{2}\sqrt{3}$.

The normalized converting factors used in fundamental cases T_{yz} and T_{zx} were $\sqrt{3}/(\sum_{j}^{J} \bar{\tau}_{yz,j}A_{y,j})$ and $1/(\sum_{i}^{I} \tau_{zx,i}A_{x,i})$ respectively.

The stresses produced for case T_{yz} are $\bar{S}_{14}, \bar{S}_{24}, \bar{S}_{34}, \bar{S}_{44}, \bar{S}_{54}$, and $\bar{S}_{64}$ ($\bar{S}_{14}=\bar{S}_{24}=\bar{S}_{34}=\bar{S}_{64}=0$), and for case T_{zx}, $\bar{S}_{15}, \bar{S}_{25}, \bar{S}_{35}, \bar{S}_{45}, \bar{S}_{55}$, and $\bar{S}_{65}$ ($\bar{S}_{15}=\bar{S}_{25}=\bar{S}_{35}=\bar{S}_{65}=0$).

TRANSVERSE SHEAR LOADING, T_{xy}

The problem of a unidirectional fiber-reinforced composite subjected to transverse shear loading is the same as fundamental cases 1 and 2, except the boundary conditions. It was found that the boundary conditions in the present case (see Appendix I) are complementary to those of fundamental transverse loading case N_x.

In addition to the study made in Appendix I on the boundary conditions, an analytical study (see Appendix II) as well as a numerical analysis (see Appendix III) by finite-element method on the symmetry relations in perforated plates has been performed. The analytical study and the numerical analysis justify the conclusions made in Appendix I.

The boundary conditions used in the computation program are:

1) Displacements in the y-direction along the boundary planes perpendicular to the x-axis are 1 and -1 respectively; i.e., V = 1 at $x = \frac{1}{2}\sqrt{3}$ and v = -1 at $x = -\frac{1}{2}\sqrt{3}$.

2) Displacements in the x-direction along the boundary planes perpendicular to the y-axis are zeroes; i.e., u = 0 at $y = \pm\frac{1}{2}$.

3) Normal stresses along the boundary planes are zeroes.

The normalized factor for converting the loading condition of the above boundary conditions into the unity loading condition was $1/(\sum_{i}^{I} \tau_{xy,i} A_{x,i})$.

The resulting stresses in this case are $\bar{S}_{16}, \bar{S}_{26}, \bar{S}_{36}, \bar{S}_{46}, \bar{S}_{56}$, and $\bar{S}_{66}$ $(\bar{S}_{46} = \bar{S}_{56} = 0)$.

PARAMETRIC STUDIES AND FINAL STRESSES IN A COMPOSITE DUE TO A GENERALIZED PLANE OBLIQUE LOADING

The coordinate system of oblique loading, which in most cases may be parallel to the surface of a unidirectional composite, will be called the loading or structural geometry system. The system with one coordinate axis parallel to the fiber axis will be called the material property system. In the previous studies of fundamental loading cases, the material property system was adopted for simplicity in description of the boundary conditions. Both systems are inclined by angle θ with their coordinate axis (Figure 2); therefore, the loadings applied on such a composite must be transformed into the loading corresponding to the material property system before the linear superposition of any desired quantitites can take place. Based on the equilibrium conditions of loading and the transformation law for rank two tensors, the three-dimensional equations of transformation have the following form:

$$N_{ij} = \ell_{ik} \ell_{j\ell} \sigma'_{k\ell} \tag{26}$$

where $\sigma'_{k\ell}$ are applied loads in psi, and N_{ij} are the transformed loadings in psi. Further, ℓ_{ik} and $\ell_{j\ell}$, the direction cosines between the new coordinate axis x y z (material property axis) (see Figure 2) and the original coordinate axis x'y'z' (loading coordinate axis), are defined in the following table.

	x'	y'	z'
x	ℓ_{11}	ℓ_{12}	ℓ_{13}
y	ℓ_{21}	ℓ_{22}	ℓ_{23}
z	ℓ_{31}	ℓ_{32}	ℓ_{33}

Equation (26) becomes after summing over k and ℓ

$$N_x = N_{11}$$

$$N_{11} = \ell_{11}\ell_{11}\sigma_{11} + \ell_{11}\ell_{12}\sigma_{12} + \ell_{11}\ell_{13}\sigma_{13} + \ell_{12}\ell_{11}\sigma_{21} + \ell_{12}\ell_{12}\sigma_{22} + \ell_{12}\ell_{13}\sigma_{23} + \ell_{13}\ell_{11}\sigma_{31} + \ell_{13}\ell_{12}\sigma_{32} + \ell_{13}\ell_{13}\sigma_{33}$$

$$T_{xy} = N_{12}$$

$$N_{12} = \ell_{11}\ell_{21}\sigma_{11} + \ell_{11}\ell_{22}\sigma_{12} + \ell_{11}\ell_{23}\sigma_{13} + \ell_{12}\ell_{21}\sigma_{21} + \ell_{12}\ell_{22}\sigma_{22} + \ell_{12}\ell_{23}\sigma_{23} + \ell_{13}\ell_{21}\sigma_{31} + \ell_{13}\ell_{22}\sigma_{32} + \ell_{13}\ell_{23}\sigma_{33}$$

$$T_{xz} = N_{13}$$

$$N_{13} = \ell_{11}\ell_{31}\sigma_{11} + \ell_{11}\ell_{32}\sigma_{12} + \ell_{11}\ell_{33}\sigma_{13} + \ell_{12}\ell_{31}\sigma_{21} + \ell_{12}\ell_{32}\sigma_{22} + \ell_{12}\ell_{33}\sigma_{23} + \ell_{13}\ell_{31}\sigma_{31} + \ell_{13}\ell_{32}\sigma_{32} + \ell_{13}\ell_{33}\sigma_{33} \qquad (27)$$

$$N_y = N_{22}$$

$$N_{22} = \ell_{21}\ell_{21}\sigma_{11} + \ell_{21}\ell_{22}\sigma_{12} + \ell_{21}\ell_{23}\sigma_{13} + \ell_{22}\ell_{21}\sigma_{21} + \ell_{22}\ell_{22}\sigma_{22} + \ell_{22}\ell_{23}\sigma_{23} + \ell_{23}\ell_{21}\sigma_{31} + \ell_{23}\ell_{22}\sigma_{32} + \ell_{23}\ell_{23}\sigma_{33}$$

$$T_{yz} = N_{23}$$

$$N_{23} = \ell_{21}\ell_{31}\sigma_{11} + \ell_{21}\ell_{32}\sigma_{12} + \ell_{21}\ell_{33}\sigma_{13} + \ell_{22}\ell_{31}\sigma_{21} + \ell_{22}\ell_{32}\sigma_{22} + \ell_{22}\ell_{33}\sigma_{23} + \ell_{23}\ell_{31}\sigma_{31} + \ell_{23}\ell_{32}\sigma_{32} + \ell_{23}\ell_{33}\sigma_{33}$$

$$N_z = N_{33}$$

$$N_{33} = \ell_{31}\ell_{31}\sigma_{11} + \ell_{31}\ell_{32}\sigma_{12} + \ell_{31}\ell_{33}\sigma_{13} + \ell_{32}\ell_{31}\sigma_{21} + \ell_{32}\ell_{32}\sigma_{22} + \ell_{32}\ell_{33}\sigma_{23} + \ell_{33}\ell_{31}\sigma_{31} + \ell_{33}\ell_{32}\sigma_{32} + \ell_{33}\ell_{33}\sigma_{33}$$

In the present analysis, only a plane oblique loading p is applied on the surface of the composite. Therefore, the transformation is two-dimensional and is simplified as (see Figure 2):

$$N_x = p \sin^2\theta \tag{28}$$

$$N_z = p \cos^2\theta \tag{29}$$

$$T_{zx} = \tfrac{1}{2} p \sin(2\theta) \tag{30}$$

$$N_y = T_{yz} = T_{xy} = 0 \tag{31}$$

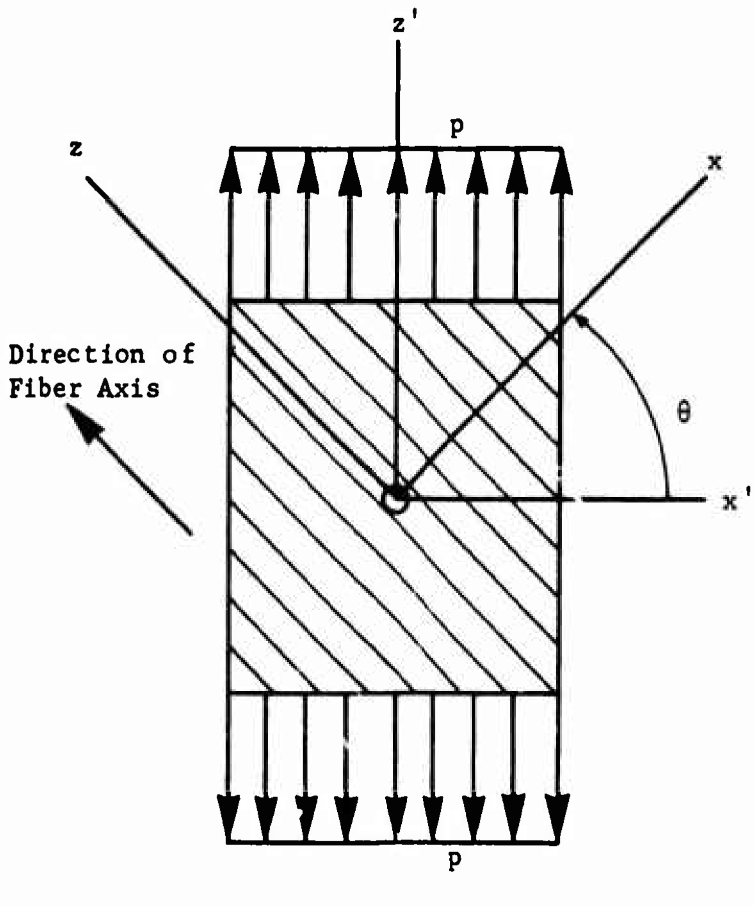


Figure 2. Loading (Geometry) Coordinate Axis and Material Property Coordinate Axes.

However, in the fundamental cases the inputs (loadings) used were $\bar{N}_x$, $\bar{N}_y, \bar{N}_z, \bar{T}_{yz}, \bar{T}_{zx}$, and $\bar{T}_{xy}$ respectively. These loadings are not completely the same as those of the left-hand side of equations (27). From the observation of the boundary conditions of fundamental cases $N_x N_y N_z$, it can be found that the internal applied loading (macrostress) on the characteristic element in each case is not one simple loading but a combination of three normal loadings (although the other two are comparatively small). In order to convert the stress components generated in these cases into those really produced by N_x, N_y, and N_z, the following manipulation must be taken to obtain the revised cases.

For revised case N_x, the simultaneous equations used to determine the constants in order to calculate stress $S_{11}, S_{21}, S_{31}, S_{41}, S_{51}$, and S_{61} due to loading N_x are:

$$1 = A_1 \sum_{i=1}^{6} (\bar{S}_{11})_i A_{x,i} + A_2 \sum_{i=1}^{6} (\bar{S}_{12})_i A_{x,i} + A_3 \sum_{i=1}^{6} (\bar{S}_{13})_i A_{x,i} \quad (32)$$

$$0 = A_1 \sum_{j=1}^{11} (\bar{S}_{21})_j A_{y,j} + A_2 \sum_{j=1}^{11} (\bar{S}_{22})_j A_{y,j} + A_3 \sum_{j=1}^{11} (\bar{S}_{23})_j A_{y,j} \quad (33)$$

$$0 = A_1 \sum_{k=1}^{158} (\bar{S}_{31})_k A_{a,k} + A_2 \sum_{k=1}^{158} (\bar{S}_{32})_k A_{z,k} + A_3 \sum_{k=1}^{158} (\bar{S}_{33})_k A_{z,k} \quad (34)$$

Solving equations (32) to (34), we get constants A_1, A_2, and A_3.

Then we can calculate the stresses produced in the revised fundamental case N_x as follows:

$$S_{11} = A_1 \bar{S}_{11} + A_2 \bar{S}_{12} + A_3 \bar{S}_{13} \quad (35)$$

$$S_{21} = A_1 \bar{S}_{21} + A_2 \bar{S}_{22} + A_3 \bar{S}_{23} \quad (36)$$

$$S_{31} = A_1 \bar{S}_{31} + A_2 \bar{S}_{32} + A_3 \bar{S}_{33} \quad (37)$$

$$S_{61} = A_1 \bar{S}_{61} + A_2 \bar{S}_{62} + A_3 \bar{S}_{63} \quad (38)$$

and $S_{41} = S_{51} = 0$. (39)

For revised fundamental case N_y, we have the simultaneous equations

$$0 = B_1 \sum_{i=1}^{6} [(\bar{S}_{11})_i A_{x,i}] + B_2 \sum_{i=1}^{6} [(\bar{S}_{12})_i A_{x,i}] + B_3 \sum_{i=1}^{6} [(\bar{S}_{13})_i A_{x,i} \tag{40}$$

$$\sqrt{3} = B_1 \sum_{j=1}^{11} [(\bar{S}_{21})_j A_{y,j}] + B_2 \sum_{j=1}^{11} [(\bar{S}_{22})_j A_{y,j}] + B_3 \sum_{j=1}^{11} [(\bar{S}_{23})_j A_{y,j} \tag{41}$$

$$0 = B_1 \sum_{k=1}^{158} [(\bar{S}_{31})_k A_{z,k}] + B_2 \sum_{k=1}^{158} [(\bar{S}_{32})_k A_{z,k}] + B_3 \sum_{k=1}^{158} [(\bar{S}_{33})_k A_{z,k} \tag{42}$$

Solving for B_1, B_2, and B_3, we can find S_{12}, S_{22}, S_{32}, and S_{62} as follows:

$$S_{12} = B_1 \bar{S}_{11} + B_2 \bar{S}_{12} + B_3 \bar{S}_{13} \tag{43}$$

$$S_{22} = B_1 \bar{S}_{21} + B_2 \bar{S}_{22} + B_3 \bar{S}_{23} \tag{44}$$

$$S_{32} = B_1 \bar{S}_{31} + B_2 \bar{S}_{32} + B_3 \bar{S}_{33} \tag{45}$$

$$S_{62} = B_1 \bar{S}_{61} + B_2 \bar{S}_{62} + B_3 \bar{S}_{63} \tag{46}$$

and

$$S_{42} = S_{52} = 0 \tag{47}$$

For revised fundamental case N_z, the simultaneous equations are:

$$0 = C_1 \sum_{i=1}^{6} [(\bar{S}_{11})_i A_{x,i}] + C_2 \sum_{i=1}^{6} [(\bar{S}_{12})_i A_{x,i}] + C_3 \sum_{i=1}^{6} [(\bar{S}_{13})_i A_{x,i}] \tag{48}$$

$$0 = C_1 \sum_{j=1}^{6} [(\bar{S}_{21})_j A_{y,j}] + C_2 \sum_{j=1}^{11} [(\bar{S}_{22})_j A_{y,j}] + C_3 \sum_{j=1}^{11} [(\bar{S}_{23})_j A_{y,j}] \tag{49}$$

$$\sqrt{3} = C_1 \sum_{k=1}^{158} [(\bar{S}_{31})_k A_{z,k}] + C_2 \sum_{k=1}^{158} [(\bar{S}_{32})_k A_{z,k}] + C_3 \sum_{k=1}^{158} [(\bar{S}_{33})_k A_{z,k}] \tag{50}$$

After solving for C_1, C_2, and C_3, we can get the stress components in the revised case N_z as follows:

$$S_{13} = C_1 \bar{S}_{11} + C_2 \bar{S}_{12} + C_3 \bar{S}_{13} \tag{51}$$

$$S_{23} = C_1 \bar{S}_{21} + C_2 \bar{S}_{22} + C_3 \bar{S}_{23} \tag{52}$$

$$S_{33} = C_1 \bar{S}_{31} + C_2 \bar{S}_{32} + C_3 \bar{S}_{33} \tag{53}$$

$$S_{63} = C_1 \bar{S}_{61} + C_2 \bar{S}_{62} + C_3 \bar{S}_{63} \tag{54}$$

and $$S_{43} = S_{53} = 0 \tag{55}$$

The stresses produced in the composite due to shears T_{yz}, T_{zx} and T_{xy} will be the same as those generated in the corresponding fundamental cases. Therefore, no revision for the shear cases is necessary. For easy identification, these stresses are reassigned as follows:

$S_{14}, S_{24}, S_{34}, S_{44}, S_{54}$ and S_{64} due to shear T_{yz},

$S_{15}, S_{25}, S_{35}, S_{45}, S_{55}$ and S_{65} due to shear T_{zx},

and $S_{16}, S_{26}, S_{36}, S_{46}, S_{56}$ and S_{66} due to shear T_{xy}.

$$(S_{14} = S_{24} = S_{34} = S_{64} = S_{15} = S_{25} = S_{35} = S_{65} = S_{46} = S_{56} = 0)$$

After obtaining the stresses in the revised fundamental cases, we then can get the final stresses by the use of the principle of linear superposition, as follows:

$$\{\sigma\} = [S]\{N\} \tag{56}$$

Written out in detail, one gets

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ T_{yz} \\ T_{zx} \\ T_{xy} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & 0 & 0 & S_{16} \\ S_{21} & S_{22} & S_{23} & 0 & 0 & S_{26} \\ S_{31} & S_{32} & S_{33} & 0 & 0 & S_{36} \\ 0 & 0 & 0 & S_{44} & S_{45} & 0 \\ 0 & 0 & 0 & S_{54} & S_{55} & 0 \\ S_{61} & S_{62} & S_{63} & 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} N_x \\ N_y \\ N_z \\ T_{yz} \\ T_{zx} \\ T_{xy} \end{Bmatrix} \tag{57}$$

FAILURE CRITERIA OF THE COMPOSITE

In the present analysis, the failure (yield) criteria were determined by examining the constituents themselves as well as their interfaces

Failure in the Constituents

The von Mises Theory of Distortion Energy was adopted as the failure criterion of the composite constituents. This theory states that the load condition $(p_{cr})_c$ is critical when the distortional energy due to

this load condition is equal to the distortional energy at failure under simple tension or compression. In mathematical form:

$$(p_{cr})_c \text{ occurs when } \frac{U_{df}}{U_d(p_{cr})} = 1$$

or

$$1 = \sqrt{2}\,\sigma_f / [(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{yz}^2 + \tau_{zx}^2 + \tau_{xy}^2)]^{\frac{1}{2}} \tag{59}$$

where σ_f is the yield strength of the material due to a simple tension or compression test and is equal to σ_{ft} when $\sigma_x + \sigma_y + \sigma_z \geq 0$ or σ_{fc} when $\sigma_x + \sigma_y + \sigma_z < 0$.

After obtaining the microstresses of each triangular prism in both constituents from the previous section, we can then find the critical loadings for all finite elements by the above equation. The smallest one will be the critical loading of the composite as far as the composite materials are concerned. However, failure may occur at the interface.

Failures at the Interface

Debonding failure at the interface between fiber and matrix is actually a very difficult "micro-micro" problem. Many investigators have attacked it on different micro-levels. Based on our survey of the literature, we found that none of them have solved it successfully in reality. Here, in our study, we have used two criteria to determine the bonding failure conditions. They are normal bonding strength and interfacial shear bonding strength between two materials.

The stress vector associated with the direction normal d at the interface is

$$[T_d] = [T][d] \tag{60}$$

or

$$[T_d] = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_y & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_z \end{bmatrix} \begin{bmatrix} \cos\xi \\ \sin\xi \\ 0 \end{bmatrix} \tag{61}$$

where Φ is the angle between direction normal at the interface and x-axis of the basic representative element.

Therefore, the normal stress of the finite elements at the interface is

$$\sigma_n = [T_d]^T[d]$$

$$= \sigma_x \cos^2\Phi + \sigma_y \sin^2\Phi + 2\tau_{xy} \sin\Phi \cos\Phi \tag{62}$$

and the tangential stress of finite elements at the interface is then

$$\tau_t = \left\{[\tfrac{1}{2}(\sigma_x - \sigma_y)\sin 2\Phi - \tau_{xy}\cos 2\Phi]^2 + (\tau_{xz}\cos\Phi + \tau_{yz}\sin\Phi)^2\right\}^{\frac{1}{2}} \tag{63}$$

One can then define each critical load condition $(p_{cr})_{in}$ and $(p_{cr})_{is}$ as that for which a developed stress is equal to a critical stress; i.e.,

$$\left(p_{cr}\right)_{in} \quad \text{occurs when} \quad \frac{\sigma_{nf}}{[\sigma_n(p_{cr})]_{av}} = \frac{\sigma_{nf}}{(\sigma_n)_{av}} = 1 \tag{64}$$

$$\left(p_{cr}\right)_{is} \quad \text{occurs when} \quad \frac{\tau_{tf}}{[\tau_t(p_{cr})]_{av}} = \frac{\tau_{tf}}{(\tau_t)_{av}} = 1 \tag{65}$$

Here $(\sigma_n)_{av}$ and $(\tau_t)_{av}$ are average values of σ_n and τ_t of neighboring triangular prisms at the interface respectively.

From equations (59), (64), and (65), we can calculate different critical loads based on the criteria of the constituents strength and the normal and tangential bonding strength of the interface. These values will be automatically compared with other stresses in the material. The smallest of these values will be the critical loading of the composite as a whole, if there is no smaller value present elsewhere in the material.

ELASTIC CONSTANTS E, G, ν, and η OF A UNIDIRECTIONAL COMPOSITE IN THE DIRECTION OF THE LOADING

For each fundamental loading case, $N_x, N_y, N_z, N_{xy}, N_{xz}, N_{yz}$, the corresponding engineering elastic constants and Poisson's ratios ν have been computed from the numerical solutions. As mentioned before, the loads acting on structures are seldom in the direction of the fibers, and they are not aligned in the direction of a Cartesian coordinate system as the fundamental loading cases are. Therefore, the elastic constants have to be calculated in the direction of the load of interest. They can be found by the following formulas[9] derived through the use of the transformation law for fourth-order tensors.

Modulus of elasticity of a composite in direction θ:

$$E_\theta = \left[\frac{m^4}{E_z} + \frac{n^4}{E_r} + \left(\frac{1}{G_{zr}} - \frac{\nu_{zr}}{E_z} - \frac{\nu_{rz}}{E_r}\right) m^2 n^2\right]^{-1} \tag{66}$$

Shear modulus of a composite in direction θ:

$$G_\theta = \left[\frac{1}{G_{zr}} + 4m^2 n^2 \left(\frac{1+\nu_{zr}}{E_z} + \frac{1+\nu_{rz}}{E_r} - \frac{1}{G_{zr}}\right)\right]^{-1} \tag{67}$$

Poisson's ratio connected with direction θ:

$$\nu_\theta = E_\theta \left[\frac{\nu_{zr}}{E_z} - m^2 n^2 \left(\frac{1+\nu_{zr}}{E_z} + \frac{1+\nu_{rz}}{E_r} - \frac{1}{G_{zr}}\right)\right] \tag{68}$$

Shear coupling factor connected with direction θ:

$$\eta_\theta = E_\theta mn \left[\frac{2m^2}{E_z} - \frac{2n^2}{E_r} + (m^2 - n^2)\left(\frac{\nu_{zr}}{E_z} + \frac{\nu_{rz}}{E_r} - \frac{1}{G_{zr}}\right)\right] \tag{69}$$

NUMERICAL RESULTS

Once the details of the stress-strain fields are known for the fundamental loading cases, the final stresses due to oblique loading can be computed by using equation (57). In this phase, the obtained stresses have been numerically introduced into the von Mises criterion equations [(59), (64), and (65)] for the finite elements shown in Figure 3.

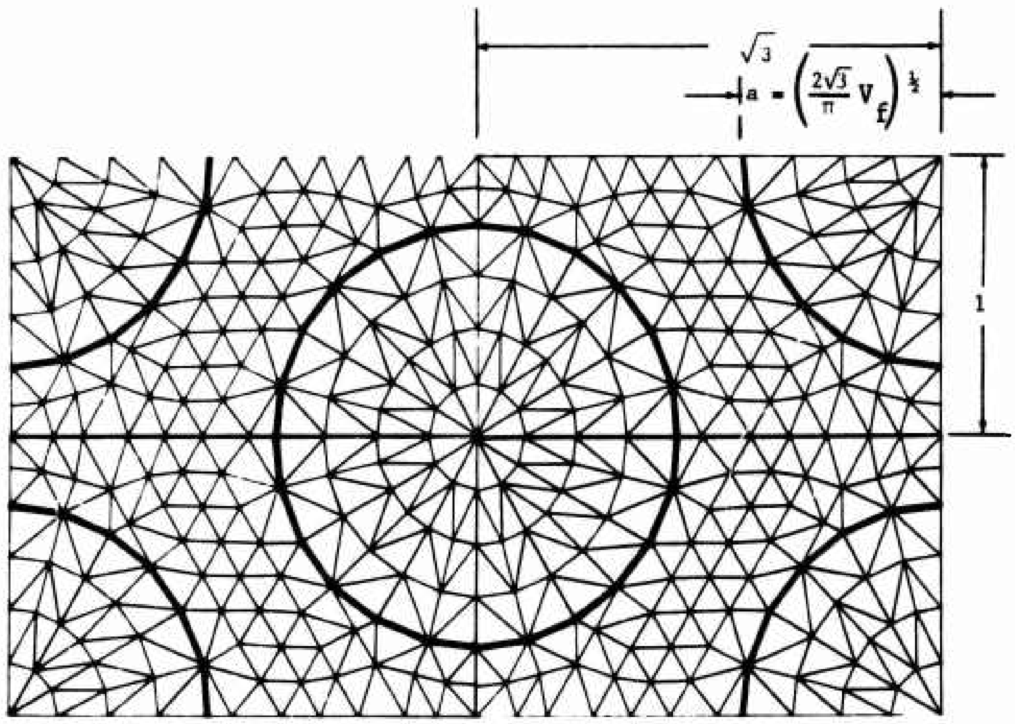


Figure 3. The Finite Elements for Which the Stresses Were Computed and the von Mises Failure Criterion Was Applied.

In this manner, the smallest critical load p_{cr}, which is the critical load of the composite, is obtained (Figure 4 through Figure 8). Equations (66) to (69) have been used to calculate the elastic constants (Figures 9 through 18), Poisson's ratio (Figure 19 through Figure 24), and shear coupling factors (Figures 25 and 26).

The different combinations of computer inputs that were used in the fundamental loading cases are for $\nu_f = 0{\cdot}2$ and $\nu_f/\nu_m = 0{\cdot}5714$ (see table below).

COMBINATIONS OF COMPONENTS, MATERIAL CONSTANTS, AND VOLUME PERCENTAGES FOR WHICH THE COMPOSITE MATERIAL CONSTANTS AND CRITICAL LOAD HAVE BEEN COMPUTED

	V_f	E_f/E_m	$E_f \cdot 10^{-6}$		V_f	E_f/E_m	$E_f \cdot 10^{-6}$
1	0	1	0·38	22	0·5	30	11·40
2	0·5	2	0·76	23	0·5	60	22·80
3	0·5	4	1·52	24	0·5	90	34·20
4	0·5	6	2·28	25	0·5	120	45·60
5	0·5	10	3·80	26	0·5	160	60·80
6	0·5	20	7·60	27	0·6	30	11·40
7	0·6	2	0·76	28	0·6	60	22·80
8	0·6	4	1·52	29	0·6	90	34·20
9	0·6	6	2·28	30	0·6	120	45·60
10	0·6	10	3·80	31	0·6	160	60·80
11	0·6	20	7·60	32	0·7	30	11·40
12	0·7	2	0·76	33	0·7	60	22·80
13	0·7	4	1·52	34	0·7	90	34·20
14	0·7	6	2·28	35	0·7	120	45·60
15	0·7	10	3·80	36	0·7	160	60·80
16	0·7	20	7·60	37	0·8	30	11·40
17	0·8	2	0·76	38	0·8	60	22·80
18	0·8	4	1·52	39	0·8	90	34·20
19	0·8	6	2·28	40	0·8	120	45·60
20	0·8	10	3·80	41	0·8	160	60·80
21	0·8	20	7·60				

In Figure 4 the composite critical strength p_{cr} is given as a function of fiber yield strength σ_f, matrix strength σ_m, component moduli E_f and E_m, and the angle θ between fiber axis and external load. A composite with 50 percent fiber volume and Poisson's ratios $\nu_f = 0.2$, $\nu_m = .35$ is assumed. The results shown are valid for both tension and compression, provided the correct ratio σ_f/σ_m is used (in some materials, this ratio may be different in tension and compression). Consider, for example, a case where $\sigma_f/\sigma_m = 25$ in tension and $\sigma_f/\sigma_m = 12.5$ in compression. Assume a load acting at an angle $\theta = 50°$ to the fiber axis. The tensile strength in this case is $p_{cr} = 0.04\ \sigma_f$. The compressive strength, on the other hand, would be $p_{cr} = 0.08\ \sigma_f$.

The dotted curves in Figure 4 indicate that for this part the critical strength has not been computed; values are plotted only for composites where $E_f/E_m > \sigma_f/\sigma_m$.

In Figure 5 the composite critical strength σ_f is presented, like in Figure 4, as function of the components properties $\sigma_f, \sigma_m, E_f, E_m$, and θ for 60 percent of fiber content by volume. Strength values are plotted for $E_f/E_m > \sigma_f/\sigma_m$ and for $E_f/E_m < \sigma_f/\sigma_m$. At the angle $\theta = 0$ or close to zero, the strength curves for $E_f/E_m \leq 90$ separate from the strength curves for values $E_f/E_m \geq 90$. This fact is indicated by an arrow for $E_f/E_m = 30$ and $E_f/E_m = 90$.

Figure 6 shows the strength of a composite, like in Figures 4 and 5, but for 70 percent of fiber content by volume. The enlarged section at the top of the figure represents the curves on an extended scale from $\theta = 0°$ to $\theta = 10°$.

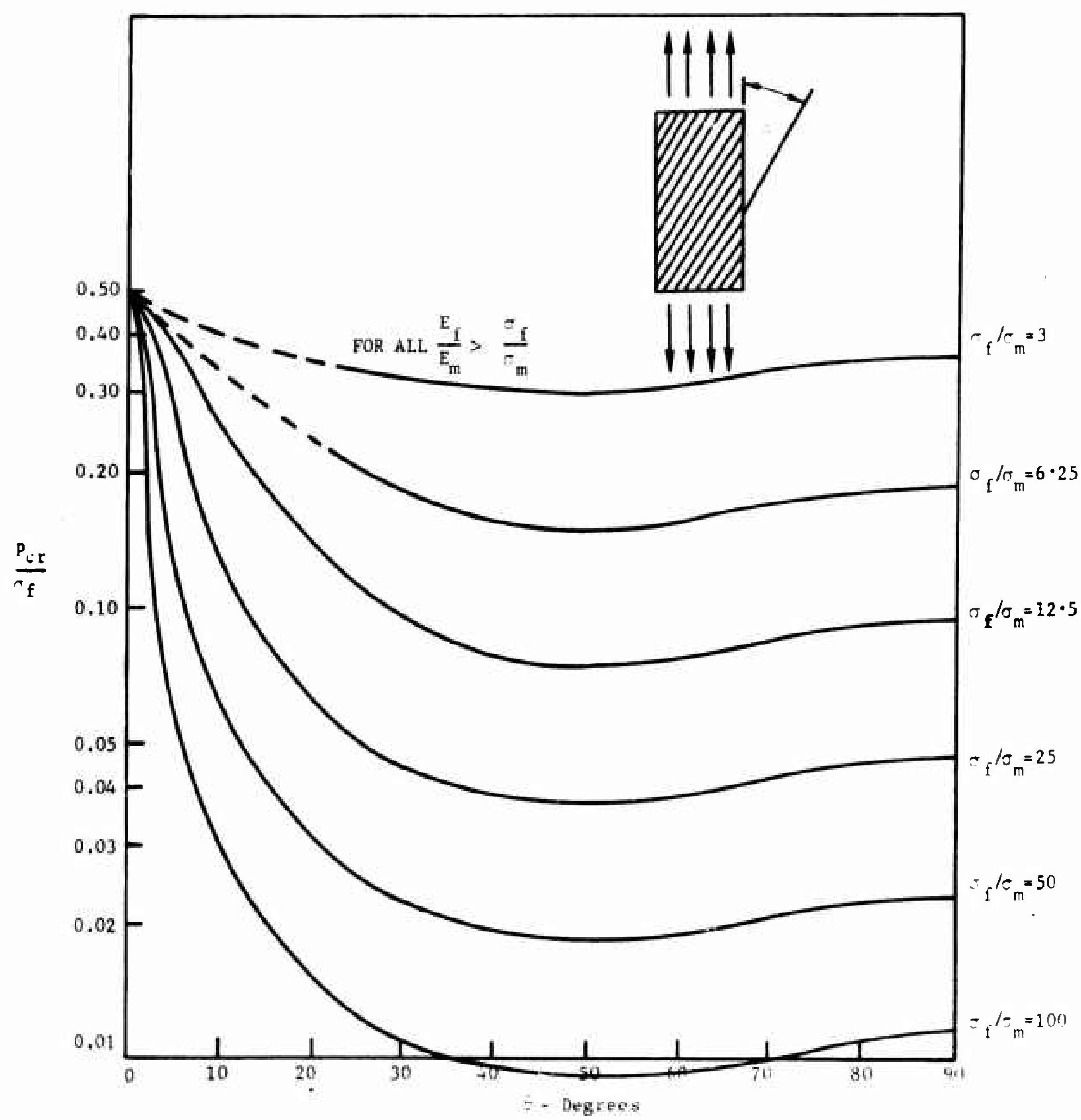


Figure 4. Critical Load (p_{cr}) of a Composite (V_f=0.5) as a Function of Loading Angle (θ), Fiber and Matrix Yield Strength, and Moduli of Elasticity (θ, σ_f, σ_m, and E_f, E_m).

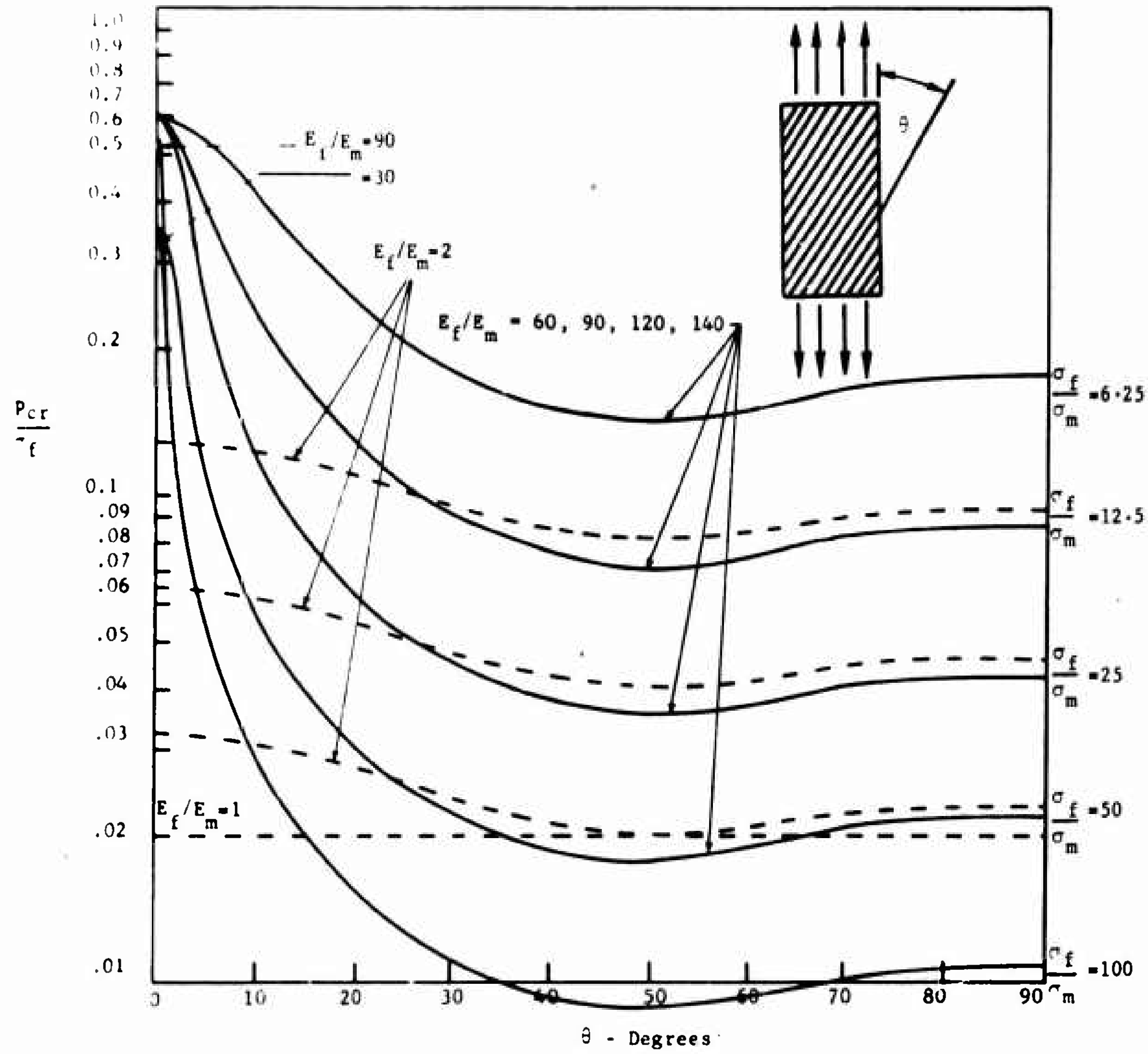


Figure 5. Critical Loading (p_{cr}) of a Composite ($V_f = 0.6$) Under Oblique Loading as a Function of Loading Angle (θ), Fiber and Matrix Yield Strength, and Moduli of Elasticity (θ, σ_f, σ_m, and E_f, E_m).

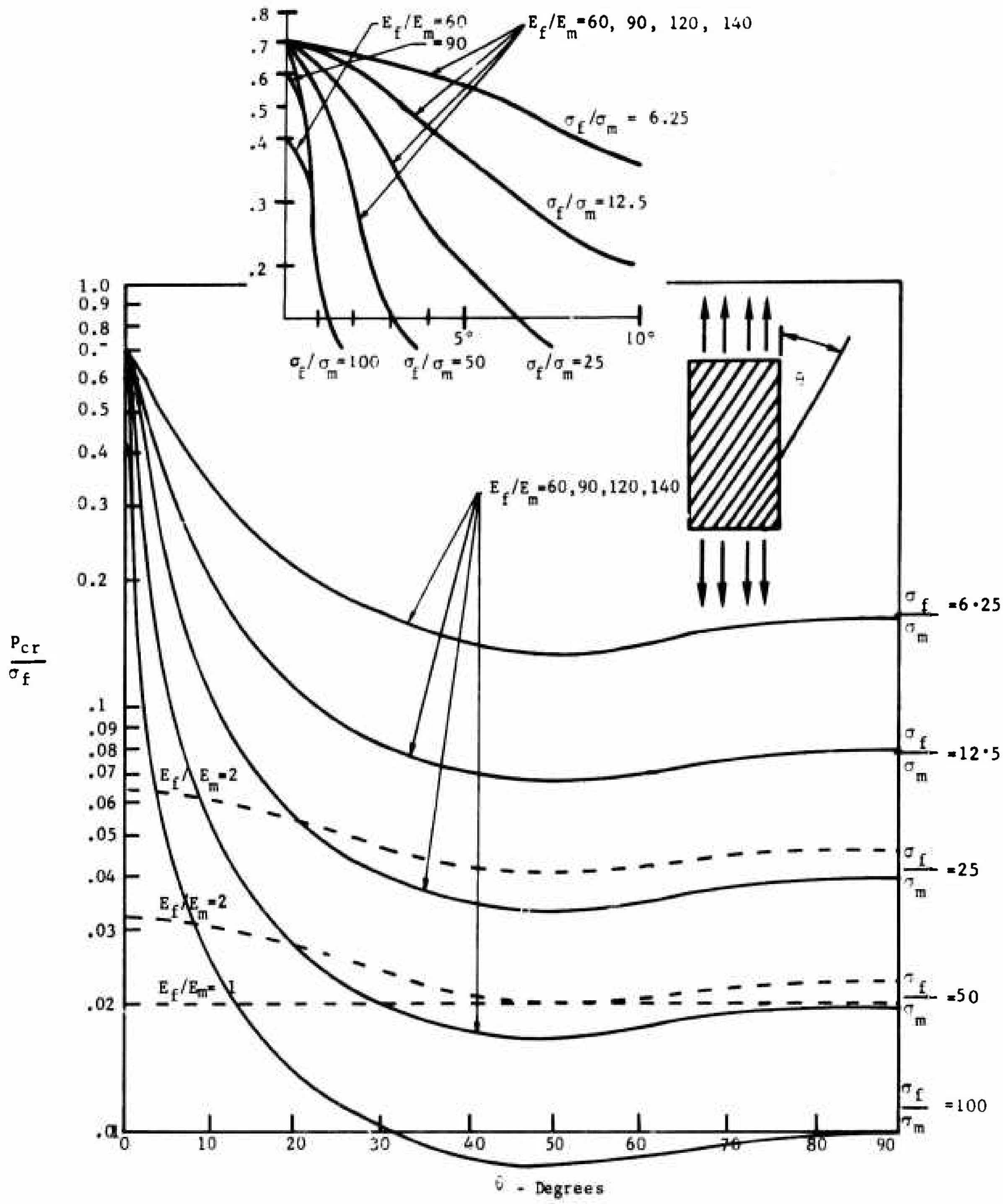


Figure 6. Critical Loading (p_{cr}) of a Composite (V_f=0.7) Under Oblique Loading as a Function of Loading Angle (θ), Fiber and Matrix Yield Strength, and Moduli of Elasticity (θ, σ_f, σ_m, and E_f, E_m).

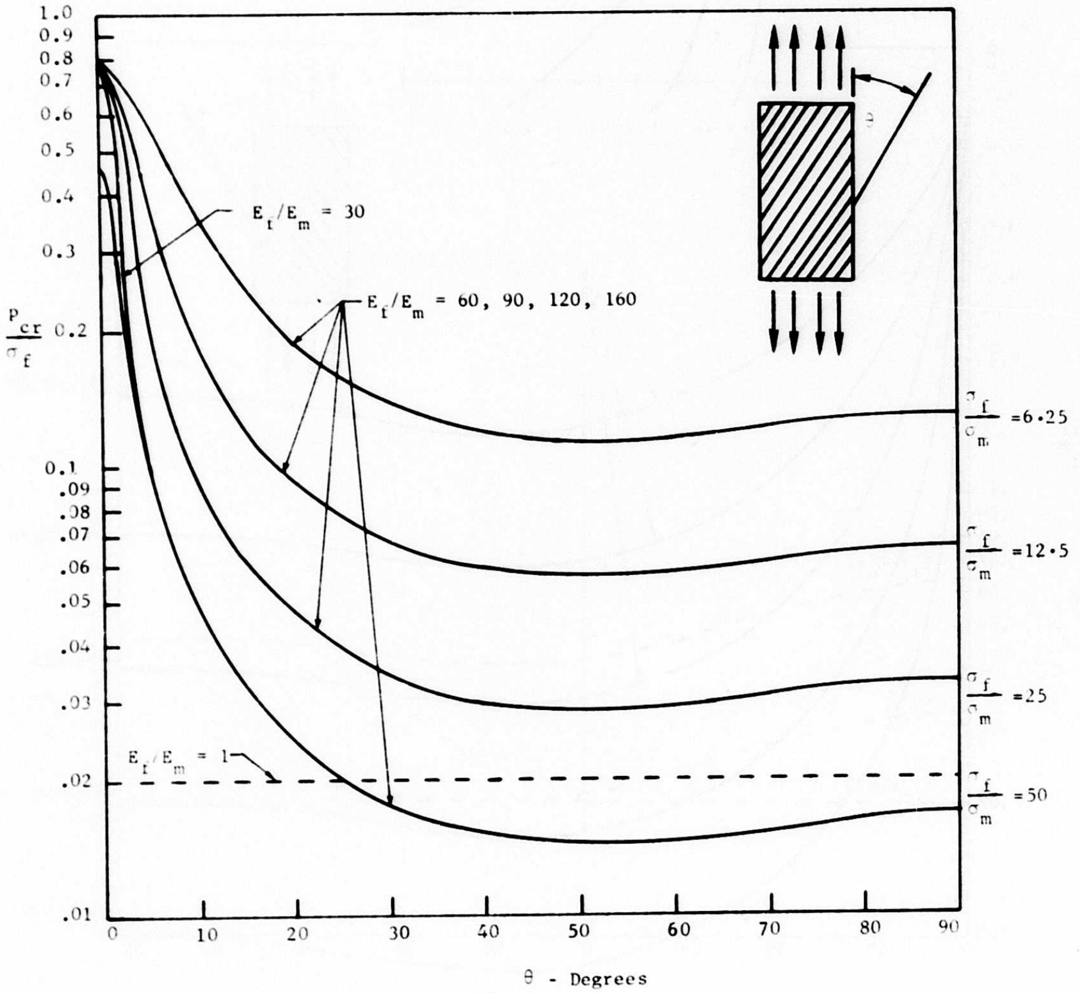


Figure 7. Critical Loading (p_{cr}) of a Composite (V_f=0.8) Under Oblique Loading as a Function of Loading Angle (θ), Fiber and Matrix Yield Strength, and Moduli of Elasticity (θ, σ_f, σ_m, and E_f, E_m).

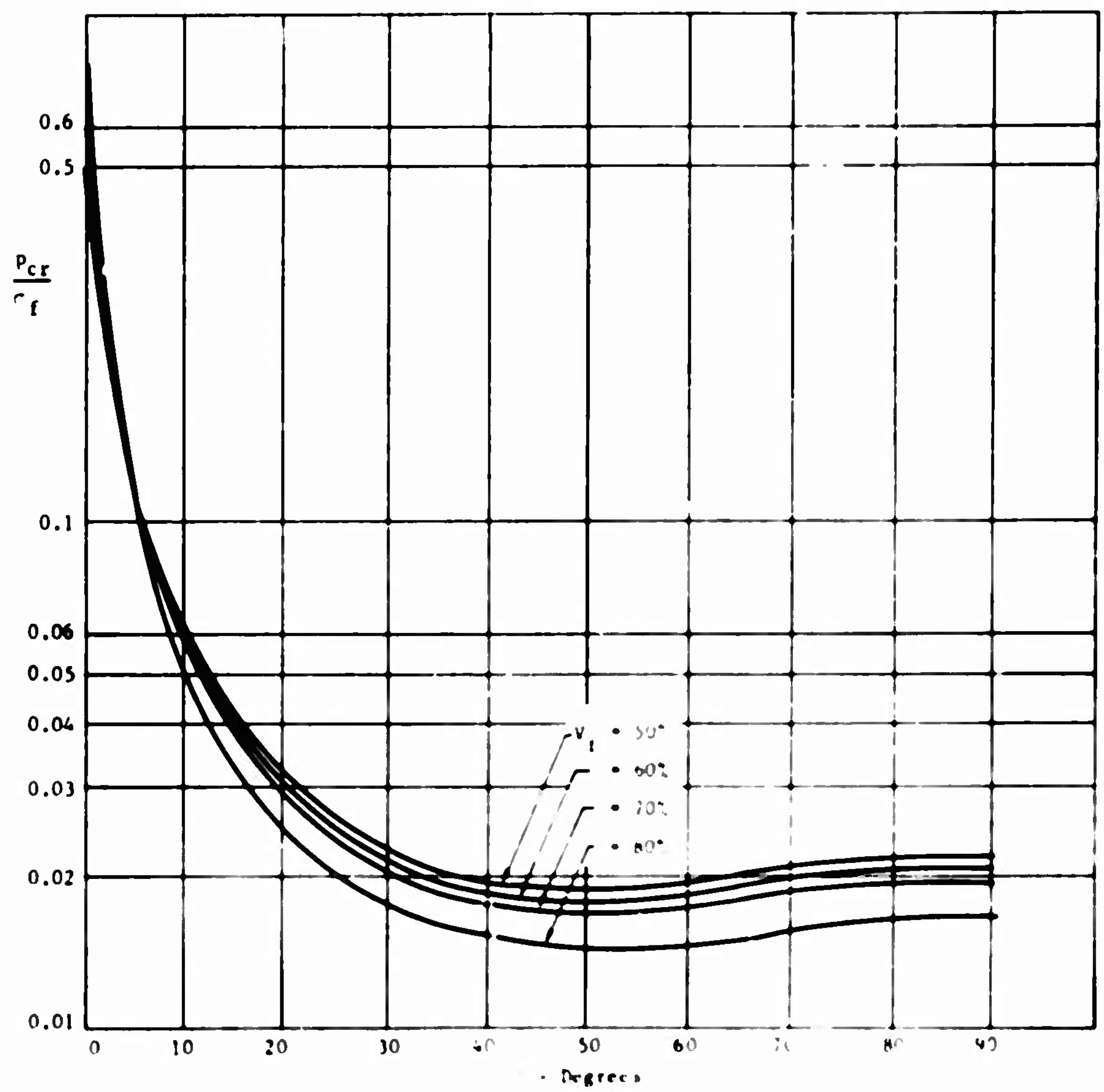


Figure 8. Critical Loading of a Composite Under Oblique Loading.

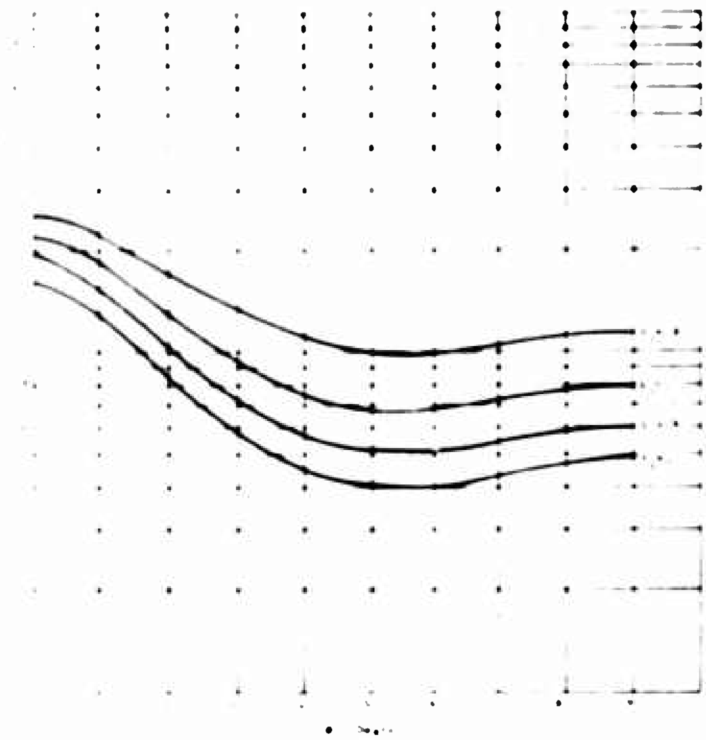


Figure 9. Major Composite - Young's Modulus vs. Fiber Orientation (E_f/E_m = 30).

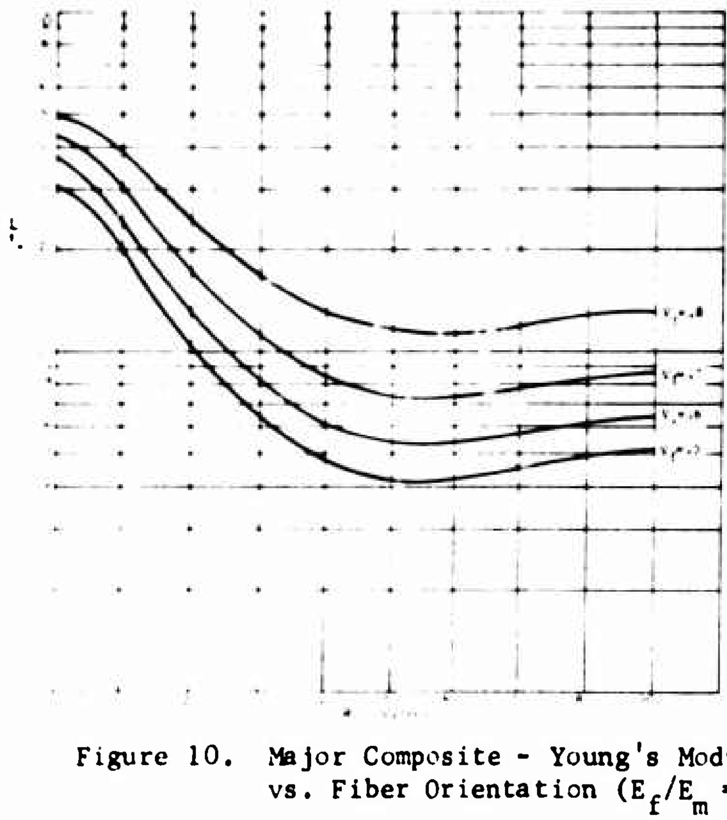


Figure 10. Major Composite - Young's Modulus vs. Fiber Orientation (E_f/E_m = 60).

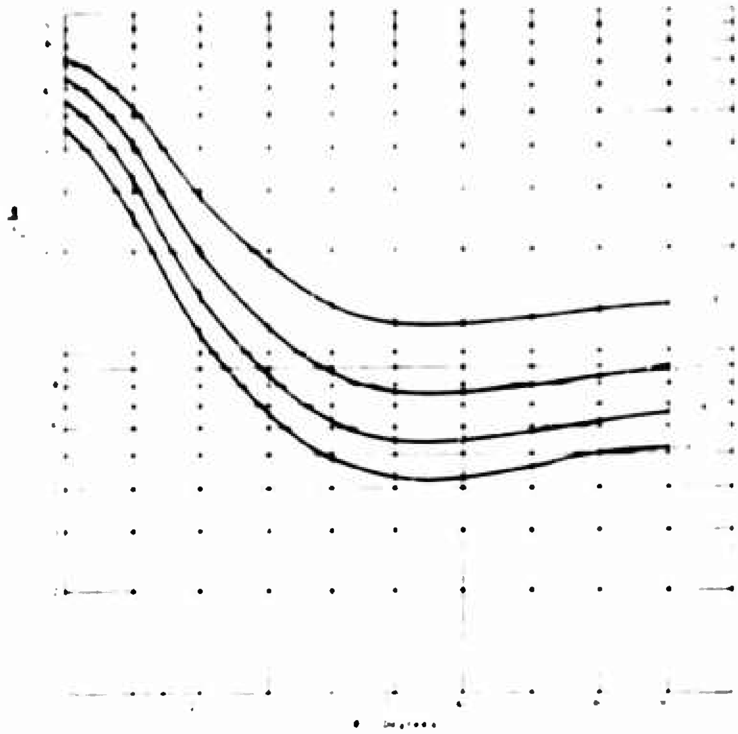


Figure 11. Major Composite - Young's Modulus vs. Fiber Orientation (E_f/E_m = 90).

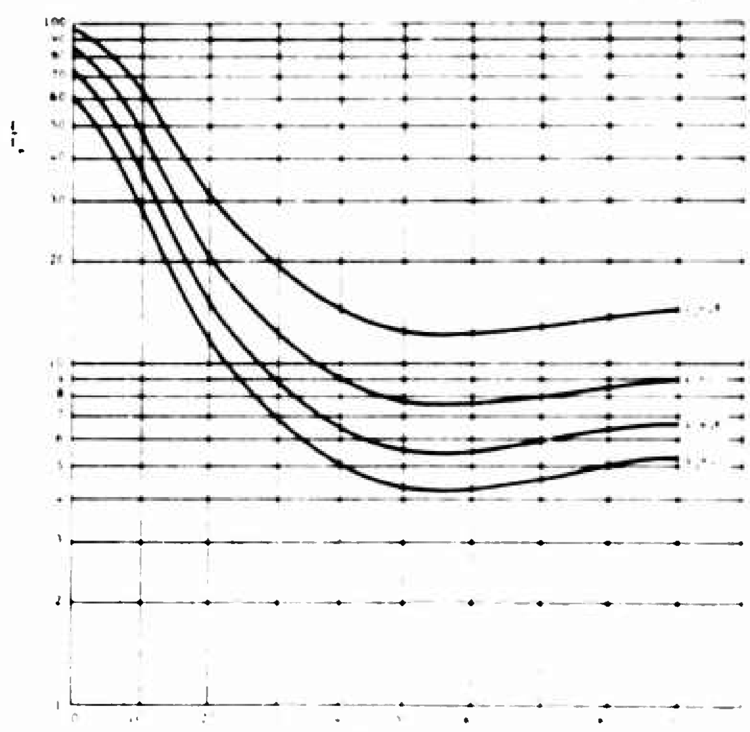


Figure 12. Major Composite - Young's Modulus vs. Fiber Orientation (E_f/E_m = 120).

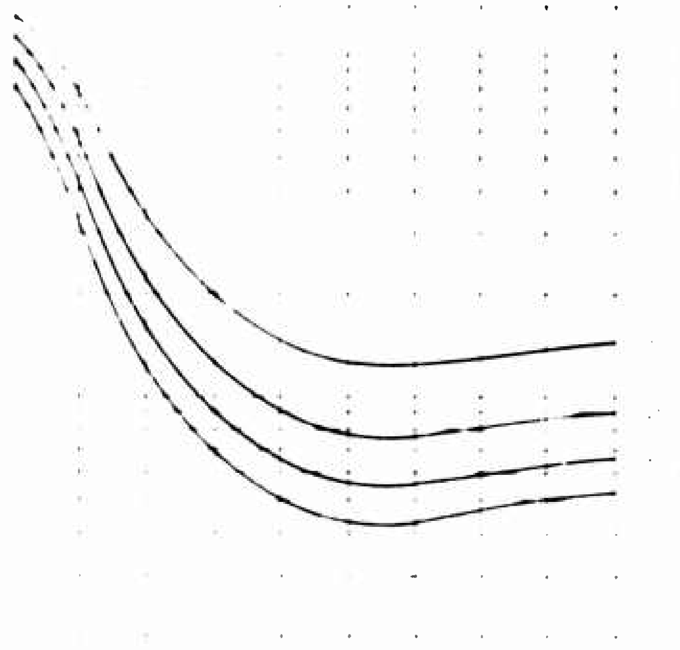


Figure 13. Major Composite - Young's Modulus vs. Fiber Orientation (E_f/E_m = 160).

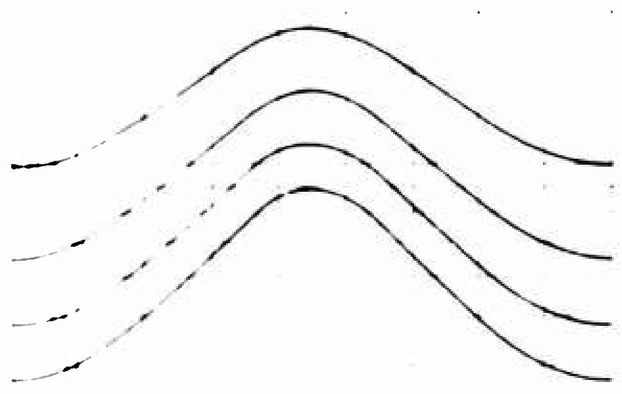


Figure 14. Composite Shear Modulus vs. Fiber Orientation (E_f/E_m = 30).

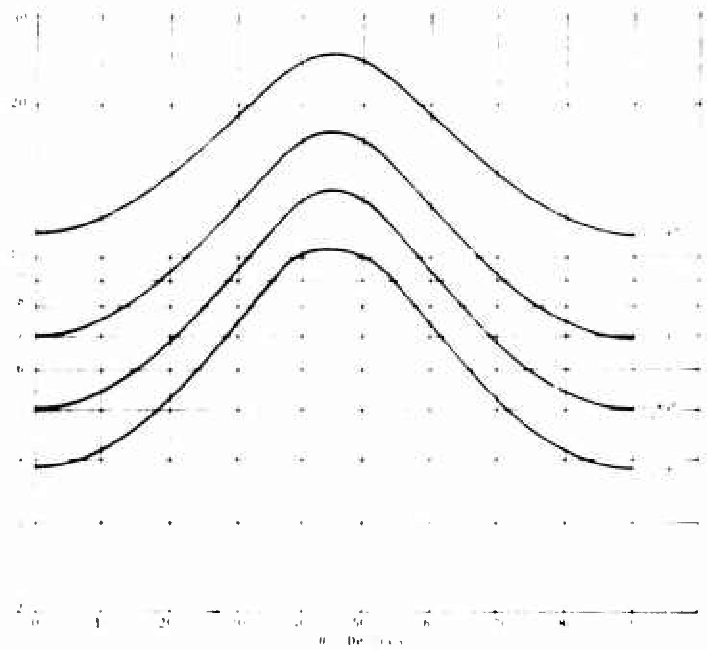


Figure 15. Composite Shear Modulus vs. Fiber Orientation ($E_f/E_m = 60$).

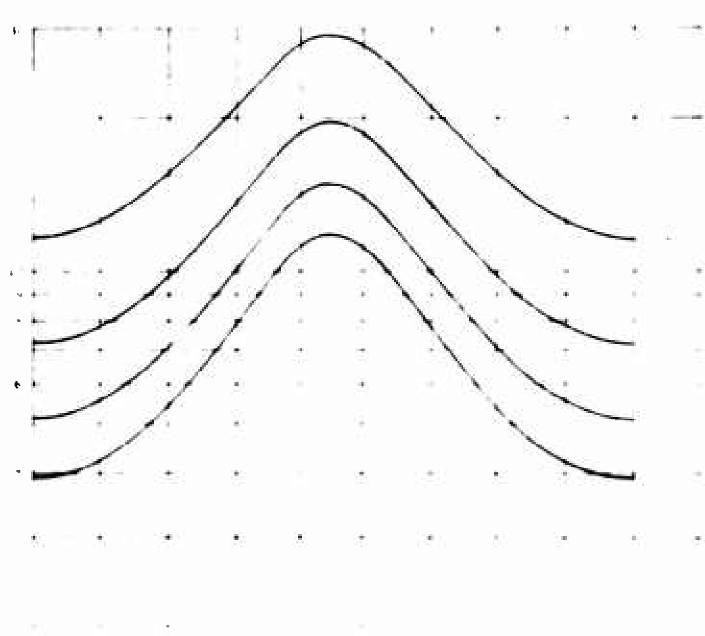


Figure 16. Composite Shear Modulus vs. Fiber Orientation ($E_f/E_m = 90$).

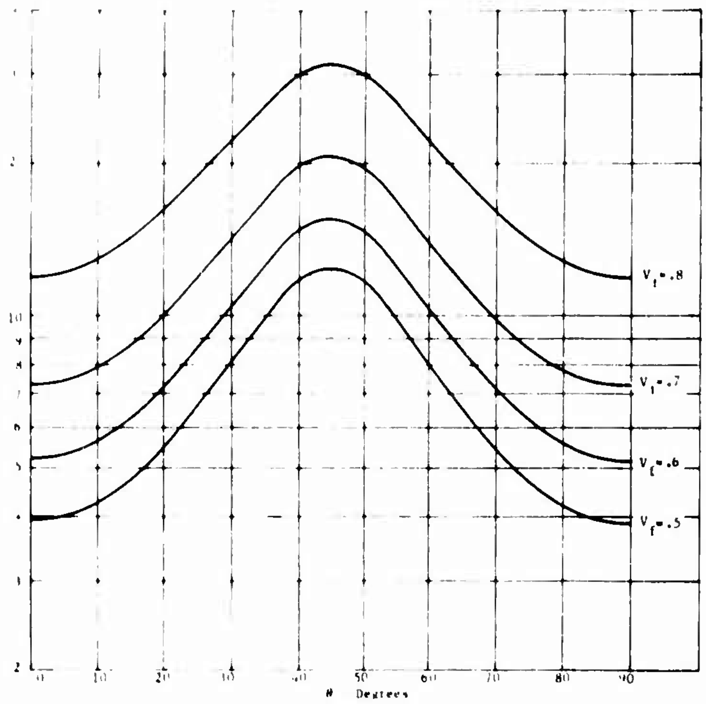


Figure 17. Composite Shear Modulus vs. Fiber Orientation ($E_f/E_m = 120$).

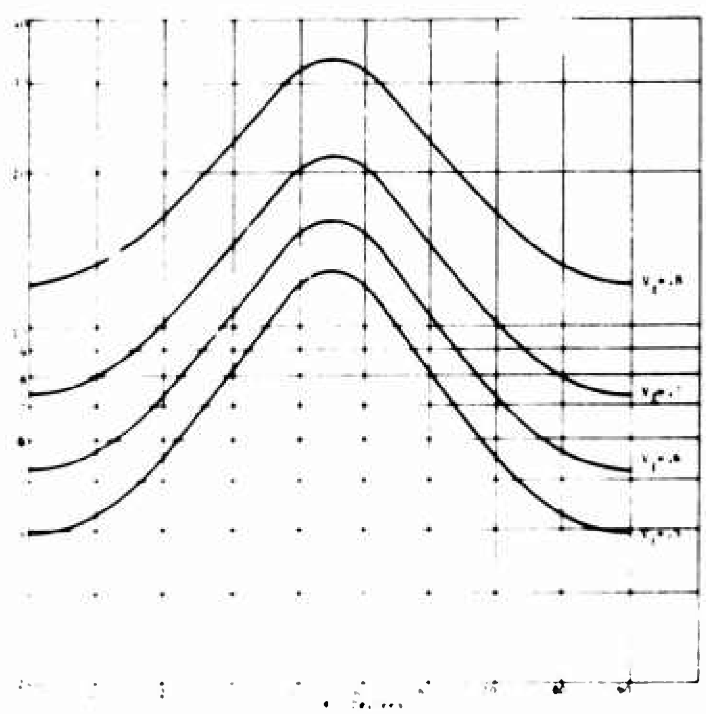


Figure 18. Composite Shear Modulus vs. Fiber Orientation ($E_f/E_m = 160$).

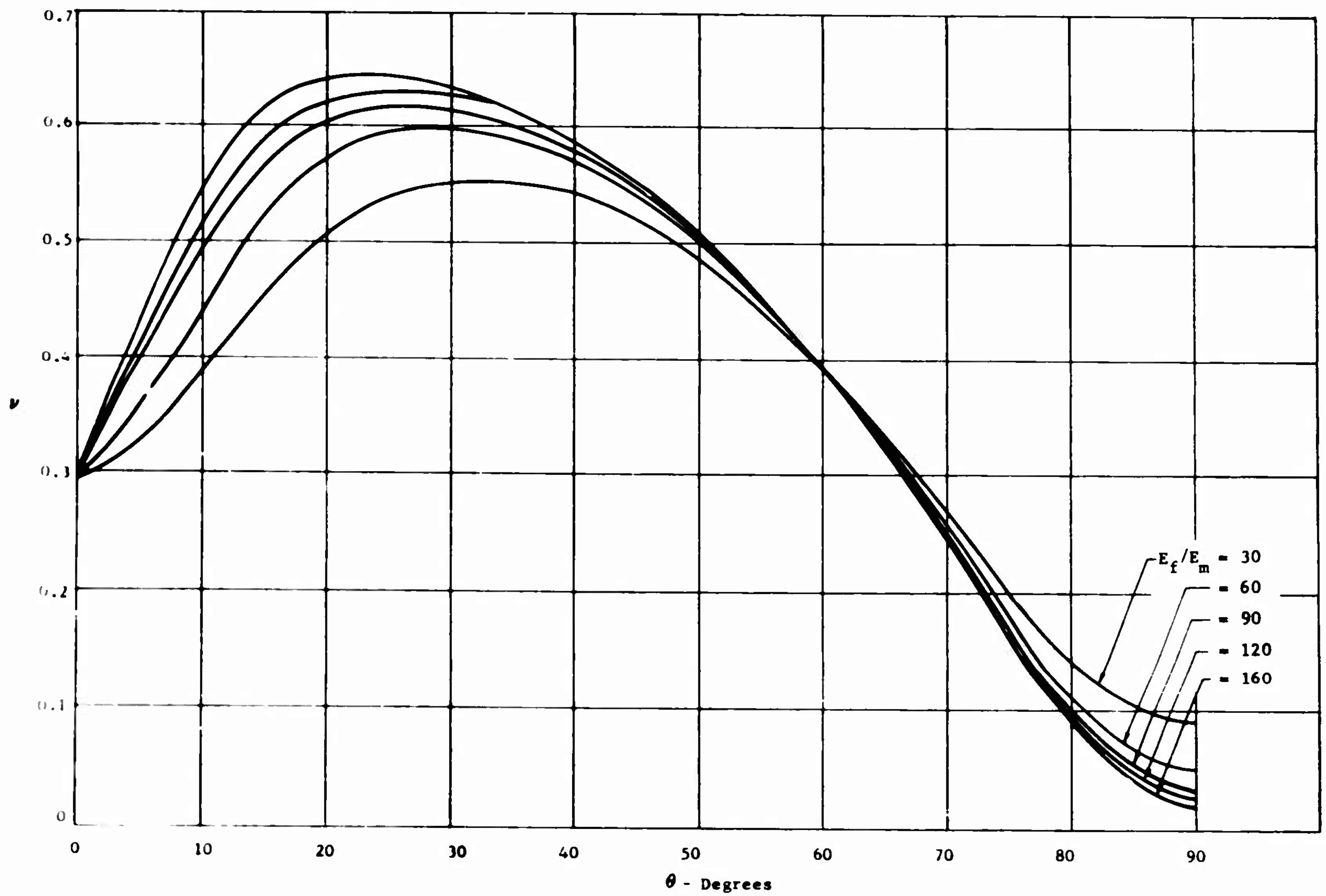


Figure 19. Major Composite - Poisson's Ratio vs. Fiber Orientation (V_f = .5).

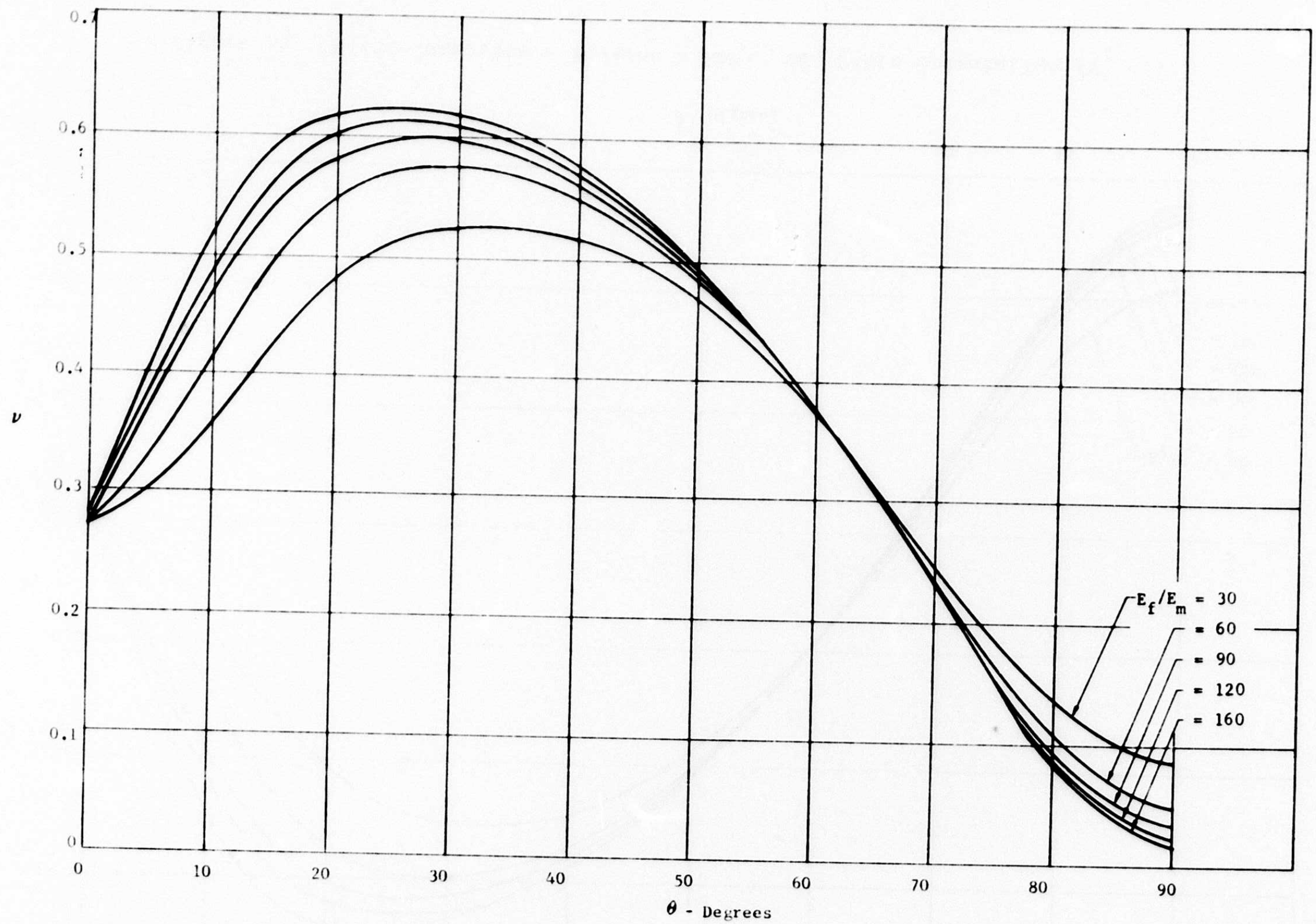


Figure 20. Major Composite - Poisson's Ratio vs. Fiber Orientation (V_f = .6).

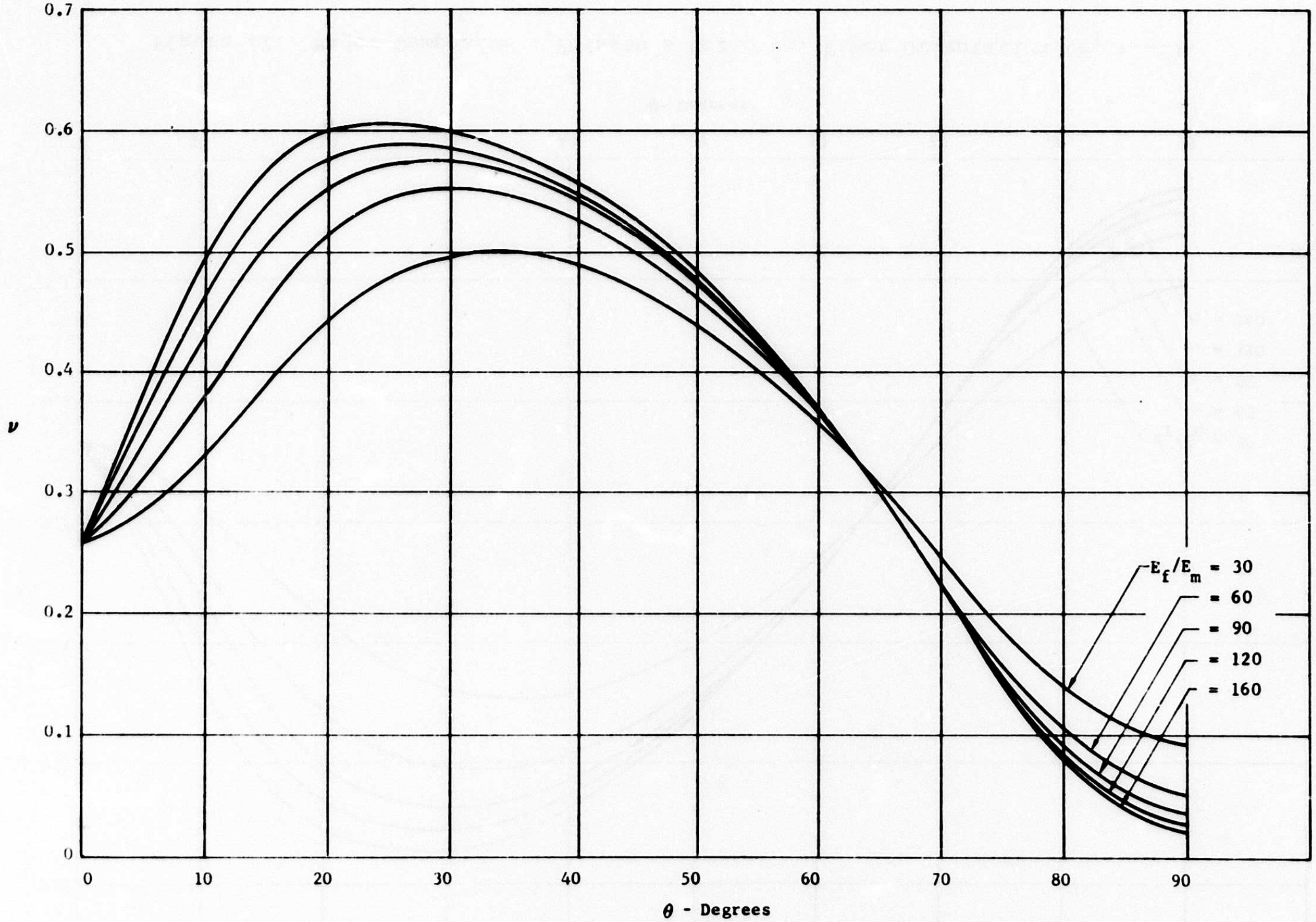


Figure 21. Major Composite - Poisson's Ratio vs. Fiber Orientation (V_f = .7).

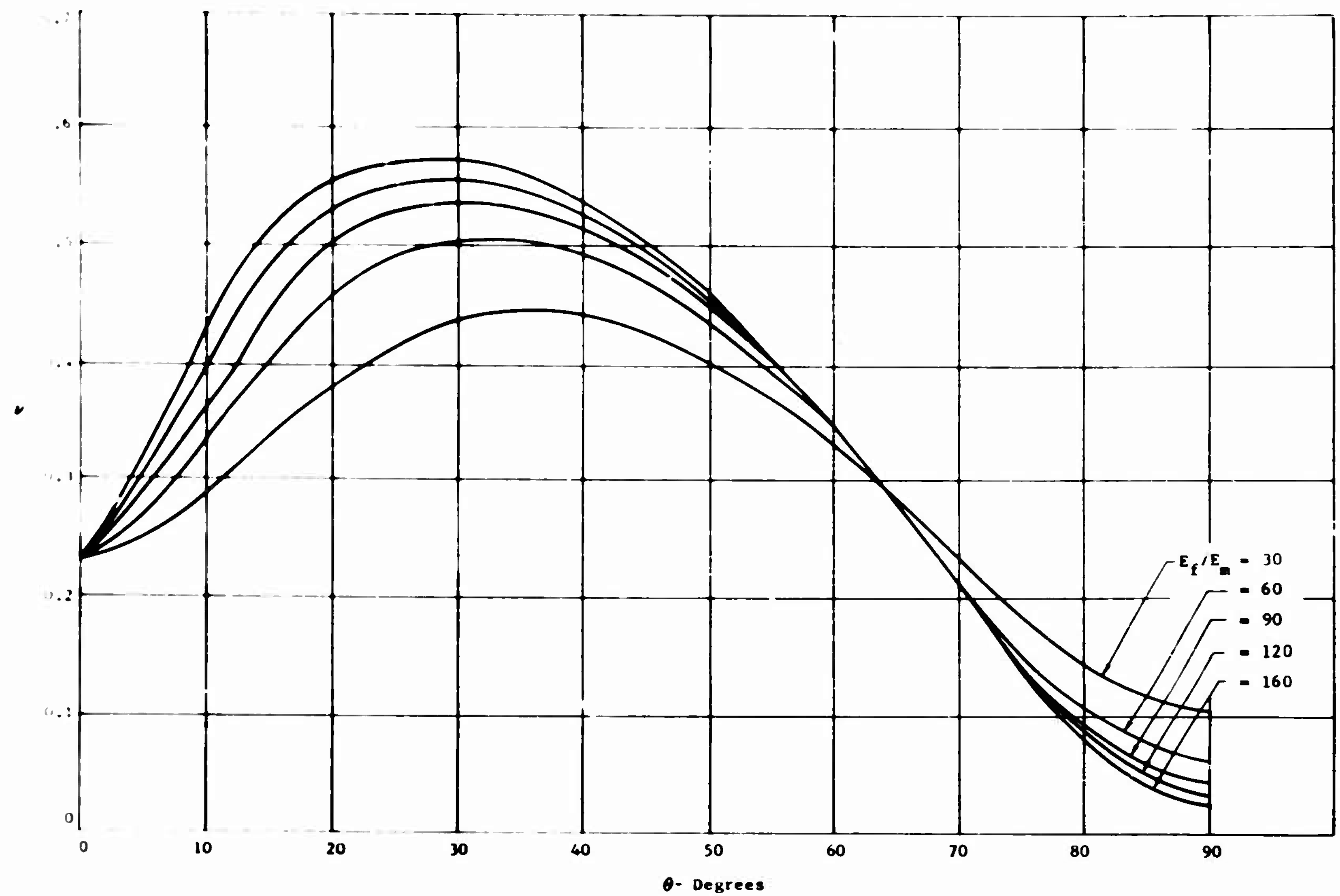


Figure 22. Major Composite - Poisson's Ratio vs. Fiber Orientation (V_f = .8).

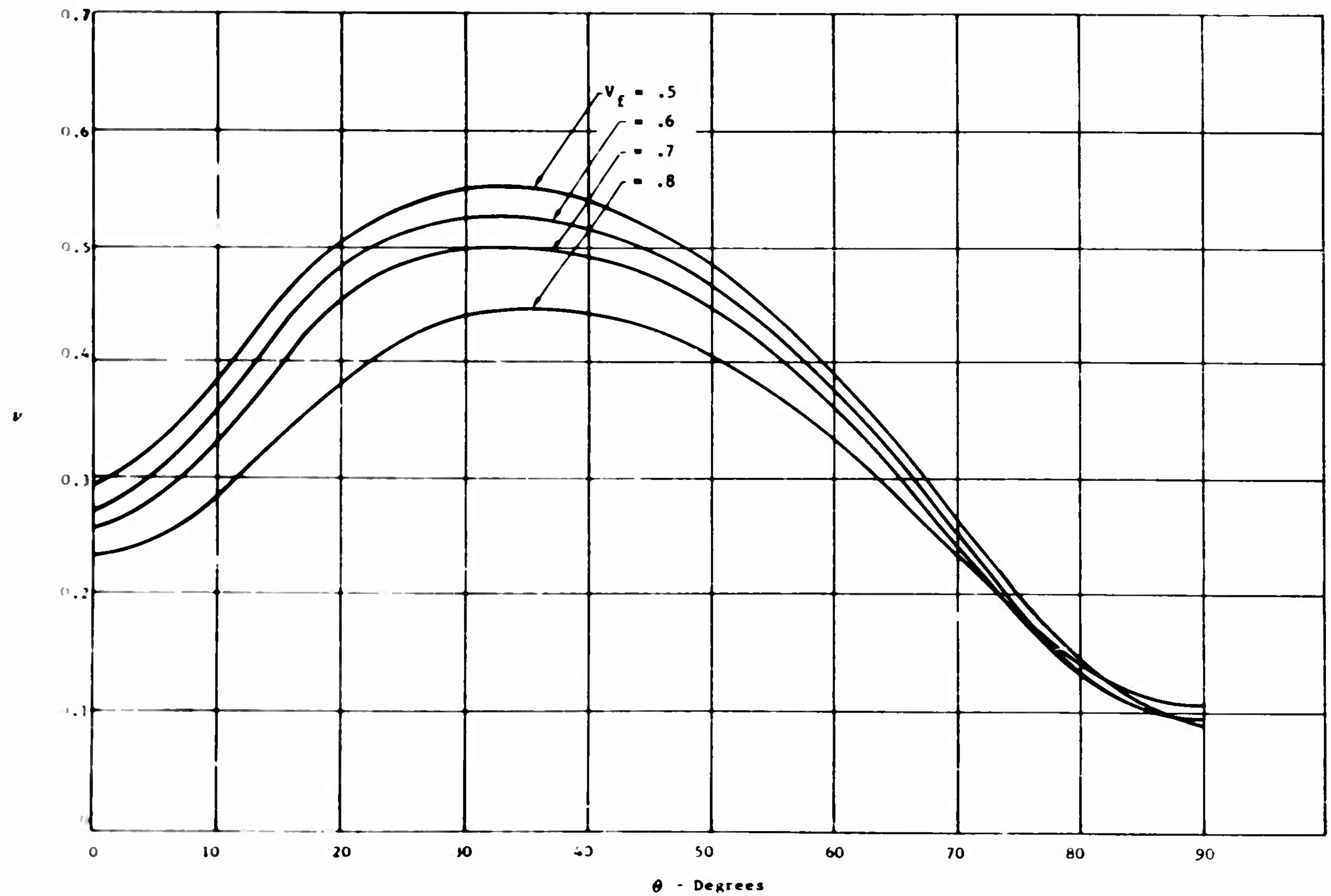


Figure 23. Major Composite - Poisson's Ratio vs. Fiber Orientation for E_f/E_m = 30.

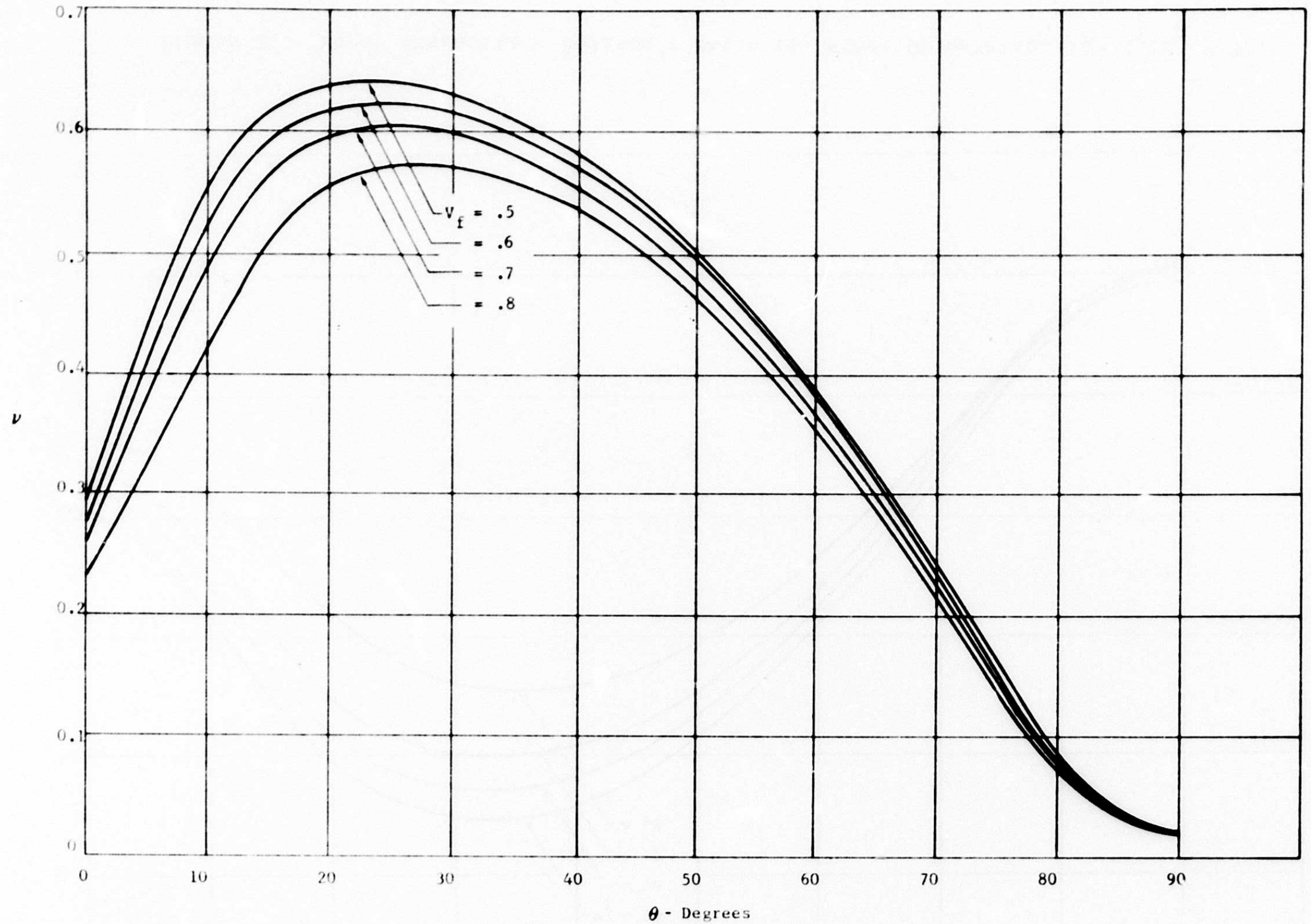


Figure 24. Major Composite - Poisson's Ratio vs. Fiber Orientation for E_f/E_m = 160.

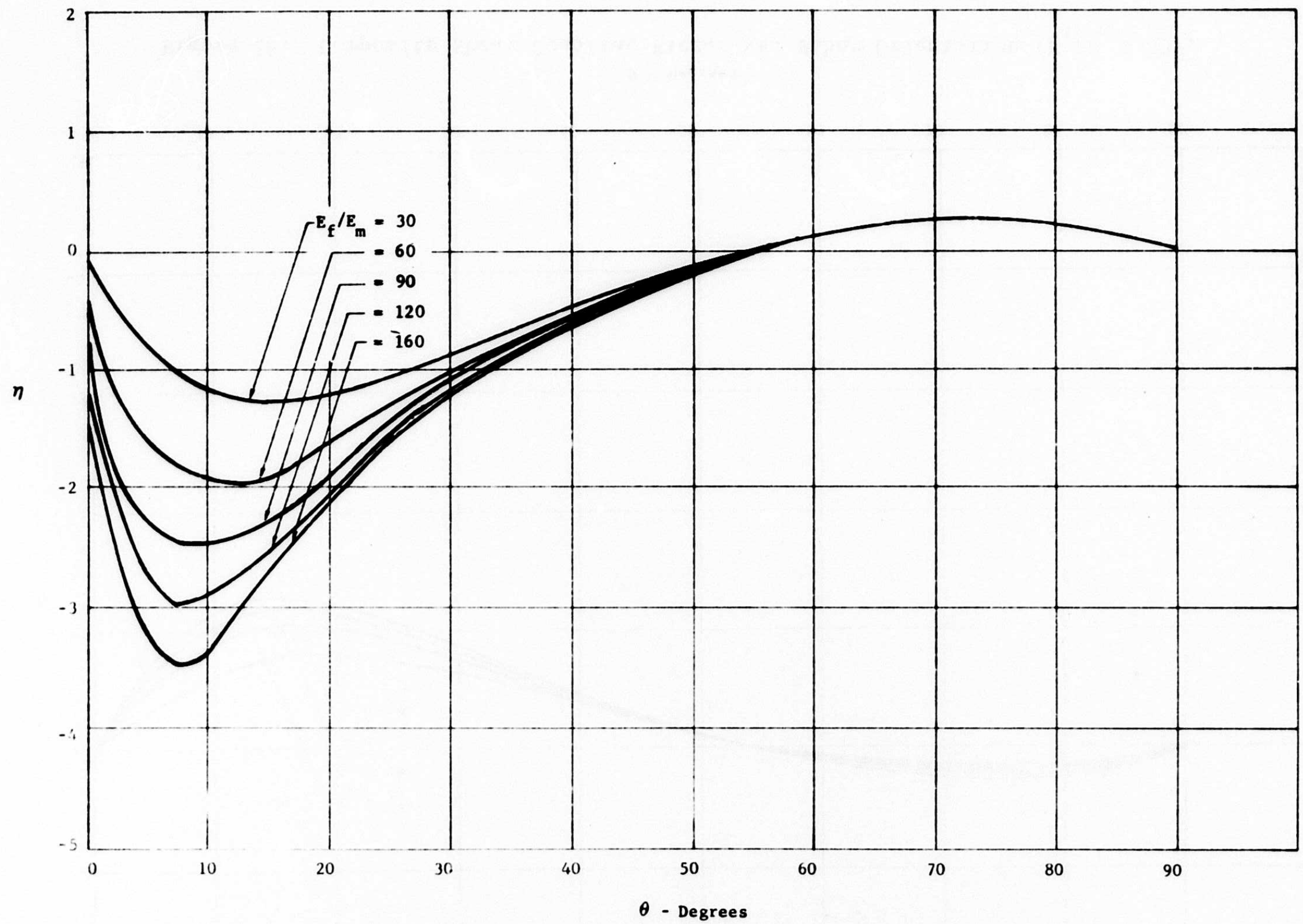


Figure 25. Composite Shear Coupling Factor vs. Fiber Orientation (V_f = .5).

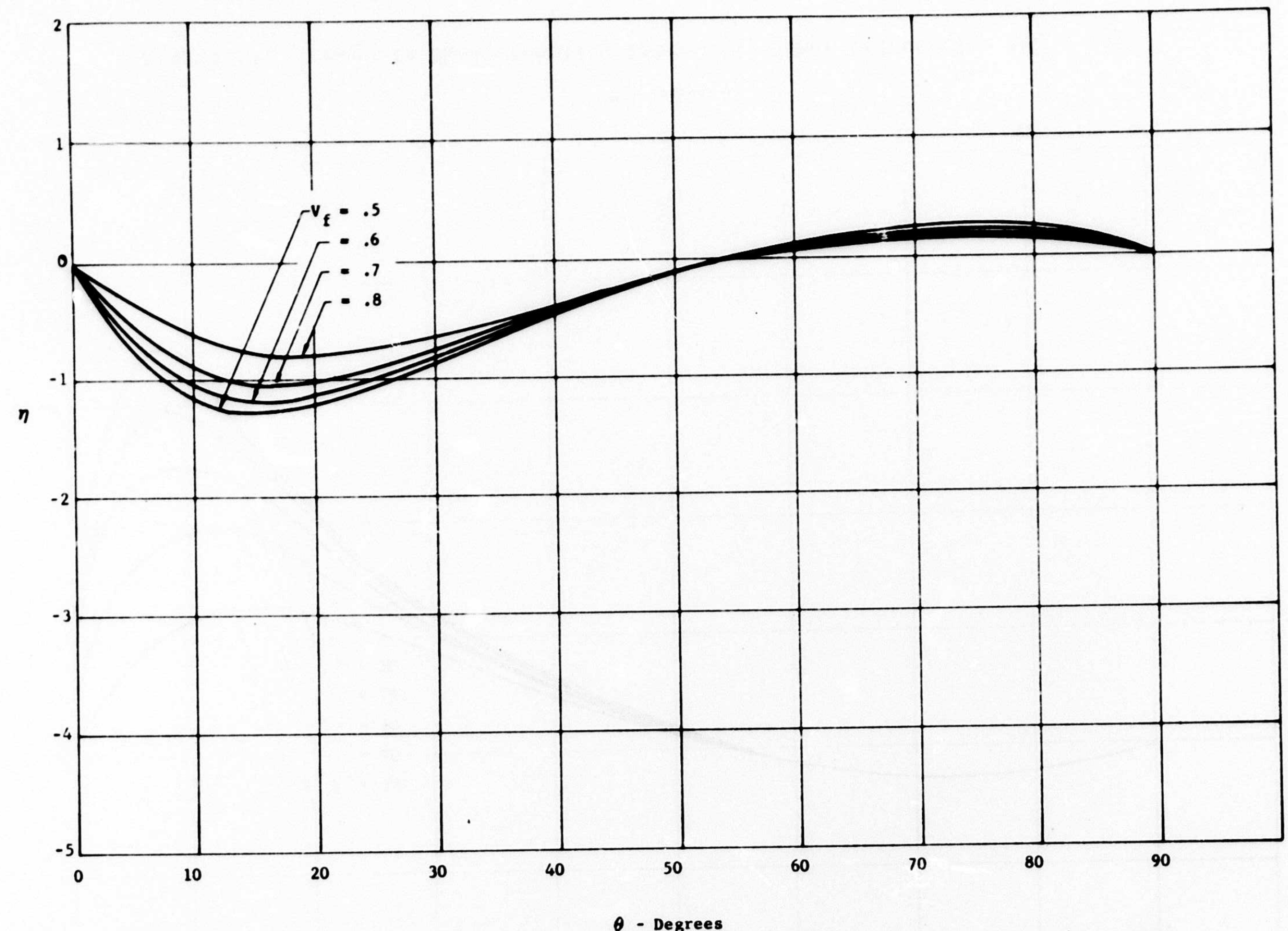


Figure 26. Composite Shear Coupling Factor vs. Fiber Orientation ($E_f/E_m = 30$).

CONCLUSIONS AND RECOMMENDATIONS

From the data presented herein and from comparisons with tests and analyses performed by other investigators (References 10, 11, and 12), it can be concluded that this kind of analysis results in valid solutions for the strength criterion of a composite. The authors of the cited papers critically compared their results with test data stemming from a variety of sources. Furthermore, results obtained by Tsai (References 13 and 14) also contribute to the confidence of the values published in this report. The comparison could be made only for axial and transverse loading because the analysis and the tests published in the literature are concerned only with these two loading conditions while the present work contains all possible analysis of loadings.

The combined stresses in each particle of the reinforcement and the matrix introduced into a strength criterion give a real picture of the strength of a composite. As can be seen from the curves, a composite loses much of its strength when the loading is not in the direction of the reinforcement. When loading in the fiber direction is present, then the failure occurs in the fiber as long as $\frac{E_f}{E_m} > \frac{\sigma_{fer}}{\sigma_{mer}}$; otherwise, failure occurs in the matrix. In cases where the loading is inclined to the fiber direction, even at very small angles, failure occurs in the resin, at the interface, and at those points where the surfaces of two adjacent fibers are closest together. The strength of a composite obtained with this analysis is on the conservative side. In reality, a composite loaded in the transverse direction, for instance, fails initially in the resin but is then capable of taking twice the load of initial failure before catastrophic structural failure occurs. The reason for this is that the matrix of most materials becomes nonlinear, and instead of failing, it smooths out the stress peaks.

An analysis which takes into account a nonlinear stress/strain relation would be the natural continuation of this work. It would reveal higher strength of a composite in the transverse direction. A realistic nonlinear analysis of a composite can be based only on micromechanics where the stress distribution in the components and the dislocations of the fibers due to loads are obtainable in detail. The present analysis possesses the above-mentioned capabilities.

LITERATURE CITED

1. Gere, J. M., STATICALLY DETERMINATE STRUCTURES, Handbooks of Engineering Mechanics, Flügge, W. (Editor), Chapter 26, p. 26-4.

2. Ashwell, D. G., NONLINEAR PROBLEMS, Handbooks of Engineering Mechanics, Flügge, W. (Editor), Chapter 45, p. 45-1.

3. Haener, J., Puppo, A., and Feng, M.-y., TRANSVERSE LOADING OF UNIDIRECTIONAL FIBER COMPOSITES, Whittaker Corporation, Research & Development Division; USAAVLABS Technical Report 69-45, U.S. Army Aviation Materiel Laboratories, Fort Eustis, Virginia, July 1969.

4. Haener, J., Ashbough, N., Chia, C.-y., and Feng, M.-y., INVESTIGATION OF MICROMECHANICAL BEHAVIOR OF FIBER REINFORCED PLASTICS, Whittaker Corporation, Research & Development Division; USAAVLABS Technical Report 67-66, U. S. Army Aviation Materiel Laboratories, Fort Eustis, Virginia, February 1968.

5. Haener, J., Puppo, A., and Feng, M.-y., OBLIQUE LOADING OF UNIDIRECTIONAL FIBER COMPOSITES: SHEAR LOADING, Whittaker Corporation, Research & Development Division; USAAVLABS Technical Report 68-81, U. S. Army Aviation Materiel Laboratories, Fort Eustis, Virginia, January 1969.

6. Adams, Donald F., and Stephen W. Tsai, THE INFLUENCE OF RADOME FILAMENT PACKING ON THE ELASTIC PROPERTIES OF COMPOSITE MATERIALS, Report RM-5608-PR, The Rand Corporation, Santa Monica, California, December 1968.

7. Wang, C.-t., APPLIED ELASTICITY, McGraw-Hill Book Company, Inc., 1953.

8. Fung, Y. C., FOUNDATIONS OF SOLID MECHANICS, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1965.

9. Hearman, R. F. S., AN INTRODUCTION TO APPLIED ANISOTROPIC ELASTICITY, Oxford University Press, 1961, pp. 10-13.

10. Noyes, J. V., and Jones, B. H., ANALYTICAL DESIGN PROCEDURES FOR THE STRENGTH AND ELASTIC PROPERTIES OF MULTILAYER FIBROUS COMPOSITES, AIAA/ASME 9th Structures Conference, Palm Springs, California, April 1-3, 1968.

11. Adams, D. F., and R. L. Thomas, TEST METHODS FOR THE DETERMINATION OF UNIDIRECTIONAL COMPOSITE SHEAR PROPERTIES, SAMPE 12th National Symposium, AC-5, 1967.

LITERATURE CITED (Continued)

12. Foye, R. L., COMPRESSION STRENGTH OF UNIDIRECTIONAL COMPOSITES, AIAA No. 66-143, 1966.

13. Tsai, S. W., STRENGTH CHARACTERISTICS OF COMPOSITE MATERIALS, NASA C7-224, Contract No. NAS 7-215, Philco Corporation, April 1965.

14. Tsai, S. W., STRUCTURAL BEHAVIOR OF COMPOSITE MATERIALS, NASA CR-71, Contract No. NAS 7-215, Philco Corporation, July 1964.

SELECTED BIBLIOGRAPHY

1. STRENGTH OF MATERIALS, PART II, ADVANCED THEORY AND PROBLEMS, D. Van Nostrand Company, Inc., New York, November 1960, pp. 444-462.

2. Murphy, G., C. E., ADVANCED MECHANICS OF MATERIALS, McGraw-Hill Book Company, Inc., 1946, pp. 70-86.

3. Broatman, L. J., and Krock, R. H., MODERN COMPOSITE MATERIALS, Addison-Wesley Publishing Company, Reading, Massachusetts, 1967.

4. Marloff, R. H., and Daniel, I. M., THREE-DIMENSIONAL PHOTOELASTIC ANALYSIS OF A FIBER REINFORCED COMPOSITE MODEL, Experimental Mechanics, April 1969, pp. 156-162.

APPENDIX I

BOUNDARY CONDITIONS FOR THE TWO-DIMENSIONAL TRANSVERSE SHEAR PROBLEM

ANALYTICAL PART

To find the stress distribution within a typical micromechanical element of a fiber-matrix configuration, proper conditions at its boundaries must be established first. The objective of this part of the analysis is to derive the boundary conditions from the geometry aspects inherent to the load configuration. In order to define the problem we refer to Figure 27, showing the assembly of micromechanical elements as they would appear within a fiber-matrix material far away from its edges and loads. Under this assumption we can consider that the stress-strain distribution around each fiber center must be identical for any arbitrarily chosen fiber center. The external loads causing the so-called "transverse shear deformation" are applied force couples, one acting in the $\pm$ x-direction and the other in the $\mp$ y-direction. In Figure 27 the forces are denoted by P_1, $-P_1$ and P_2, $-P_2$, which we parallel to the x- and y-axes, respectively. In general, magnitudes of the force couples are independent of each other, but are usually the results of some equilibrium conditions.

In order to find the boundary conditions for the rectangle ABCD, we must also consider the two adjacent rectangles A'DCB' and CBA"D" and investigate how the displacement vectors in these rectangles can be related to each other. For this purpose, we introduce three local Cartesian coordinate systems with their origins at the rectangle centroids M_o, M_o' and M_o''. The above-defined coordinate systems are the x, y - system, the x'y' - system, and the x", y" - system. The two latter coordinate axes are generated from the former by translation; i.e.,

$$x' = x \qquad y' = y - b \tag{70}$$

$$x'' = x - 2c \qquad y'' = y \tag{71}$$

Let us now consider a point P within the boundaries of rectangle ABCD which has coordinates (x, y) with respect to the non-primed coordinate system. We denote the displacement vector at P by $\bar{u}(x,y)$, which can be expressed by its components as follows:

$$\bar{u}(x,y) = \bar{i}\, u_x(x,y) + \bar{j}\, u_y(x,y) \tag{72}$$

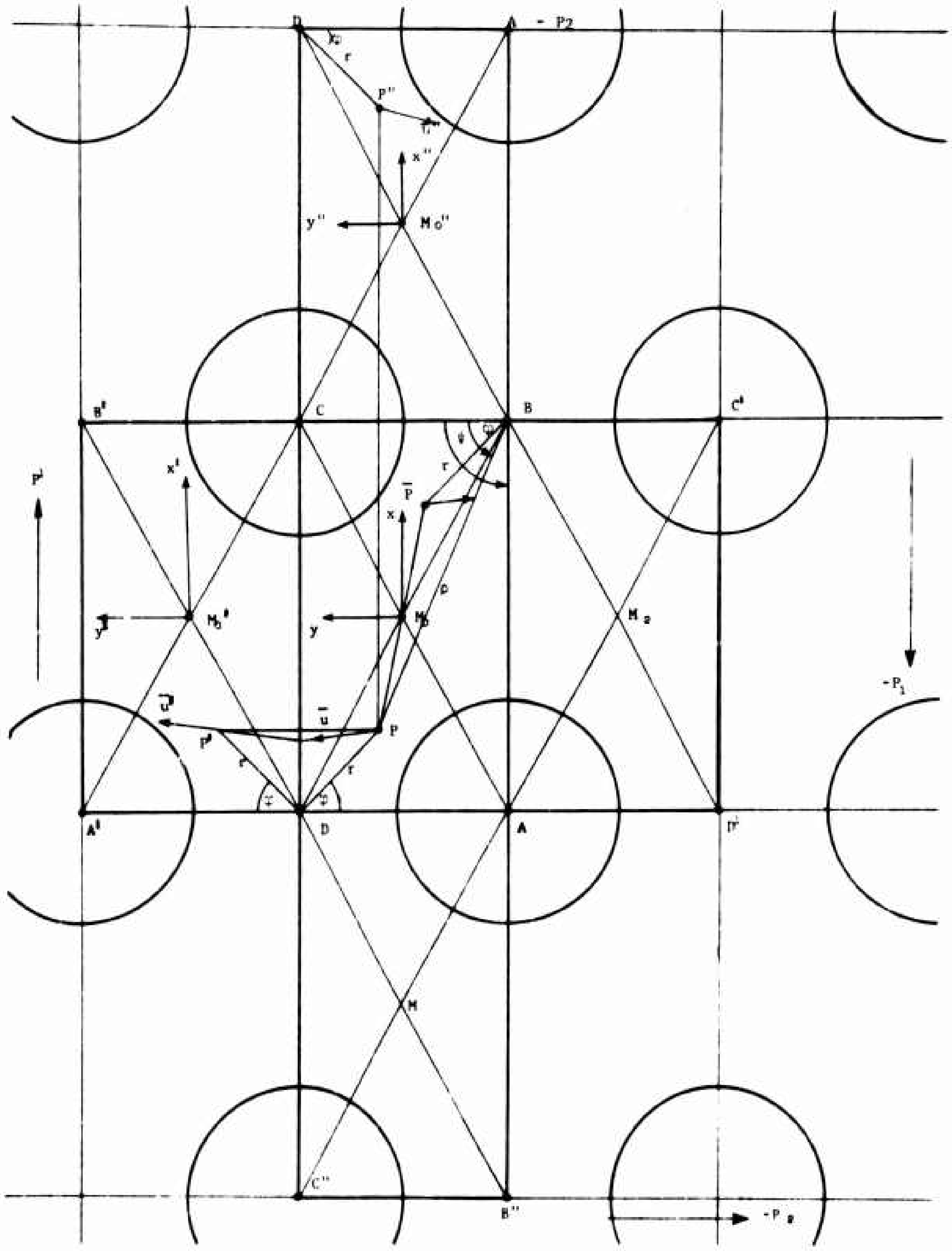


Figure 27. Geometry of the Basic Representative Element.

The displacement $\bar{u}(x,y)$ is in reference to point M_o, and we note that

$$\bar{u}(0,0) = 0$$

and therefore

$$u_x(0,0) = 0 \qquad u_y(0,0) = 0 \tag{73}$$

If we now look at the rectangle upside down, then we notice that the configuration as well as external force geometry remains unchanged. That means that the displacement vector at the point $\bar{P}(-x,-y)$ in the 180° rotated coordinate system must be identical to the displacement vector at $P(x,y)$. Therefore,

$$\bar{i}\ u_x(x,y) + \bar{j}\ u_y(x,y) = -\ \bar{i}\ u_x(-x,-y) - \bar{j}\ u_y(-x,-y)$$

or

$$u_x(x,y) = -\ u_x(-x,-y) \qquad u_y(x,y) = -\ u_y(-x,-y) \tag{74}$$

Equation (74) shows the fact that within each rectangle there exists a pair of points which are located centrally symmetrical with respect to the centroid and for which the relative displacements are also centrally symmetrical with respect to each other.

Next we have to consider rectangle A'DCB' and investigate the relative displacement vector with respect to centroid M_o' at a point $P'(x',y')$ within that rectangle. The relative displacement at P' is

$$\bar{u}'(x',y') = \bar{i}\ u_x'(x',y') + \bar{j}\ u_y'(x',y') \tag{75}$$

At the point $M_o'(x'=0,y'=0)$, we have

$$u_x'(0,0) = 0 \qquad u_y'(0,0) = 0 \tag{76}$$

The fiber configuration within rectangle A'DCB' is the mirror image of the fiber configuration within rectangle ABCD with the mirror line $y' = -\frac{b}{2}$ (or $y = +\frac{b}{2}$). However, the mirror images of the external force configuration become antidirectional but collinear. This means that by changing

the sign of all external forces, the relative displacement vector in rectangle A'DCB' will be the mirror image of the relative displacement vector within rectangle ABCD. The mirror line is again $y' = -\frac{b}{2}$. Changing the sign of external force for rectangle A'DCB' is equivalent to changing the sign of the displacement vector. Let us consider point $P'(x_M, y_M)$ being the mirror image of point $P(x,y)$ with respect to $y = +\frac{b}{2}$; then

$$x_M = x$$

$$y_M = y + 2\left(\frac{b}{2} - y\right) = b - y$$

or with (70),

$$\begin{aligned} x'_M &= x \\ y'_M &= y_M - b = b - y - b = -y \end{aligned} \tag{77}$$

The negative mirror image of vector $\bar{u}(x,y)$ will be $\bar{u}'(x'_M, y'_M)$ as follows:

$$\bar{u}'(x'_M, y'_M) = -\left\{\bar{i}\, u_x(x,y) - \bar{j}\, u_y(x,y)\right\} \tag{78}$$

Because of (77) and $\bar{i} = \bar{i}'$, $\bar{j} = \bar{j}'$,

$$u'(x'_M, y'_M) = \bar{i}\, u'_x(x,-y) + \bar{j}\, u'_y(x,-y) \tag{79}$$

Combination of (78) and (79) gives

$$\bar{i}\left[u'_x(x,-y) + u_x(x,y)\right] + \bar{j}\left[u'_y(x,-y) - u_y(x,y)\right] = 0 \tag{80}$$

This vector equation can only be satisfied if each component becomes zero. Therefore, the following two equations are obtained:

$$u'_x(x,-y) = -u_x(x,y) \tag{81}$$

$$u'_y(x,-y) = u_y(x,y) \tag{82}$$

Equations (81) and (82) show the relative displacements in rectangles A'DCB' and ABCD to each other.

In a similar fashion we can get a relation between the relative displacements in rectangle BA"D"C and the corresponding relative displacements in rectangle ABCD. We make use of the mirror symmetry about the line $x = c$ and apply the same line of reasoning as before (to distinguish from other symmetries, the sub-index M is primed).

The relative displacement at a point P" (x",y") within rectangle BA"D"C is

$$\bar{u}''(x'',y'') = \bar{i}''u_x''(x'',y'') + \bar{j}''u_y''(x'',y'') \tag{83}$$

Since $\bar{u}''(x'',y'')$ is in reference to M_o'', it is again

$$u''(0,0) = 0$$

or

$$u_x''(0,0) = 0 \text{ and } u_y''(0,0) = 0 \tag{84}$$

The mirror symmetric point to P with respect to line $x = c$ has the coordinates $x_{M'}$, $y_{M'}$ as follows:

$$x_{M'} = x + 2(c - x) = 2c - x$$

$$y_{M'} = y$$

or with (71),

$$x''_{M'} = - x; \; y''_{M'} = y \tag{85}$$

We consider the negative mirror image of vector $\bar{u}(x,y)$ with respect to line $x = c$ and obtain

$$\bar{u}''(x''_{M'}, y''_{M'}) = - \left\{- \bar{i}\, u_x(x,y) + \bar{j}\, u_y(x,y)\right\} \tag{86}$$

Because of (83) and $\bar{i} = \bar{i}''$, $\bar{j} = \bar{j}''$, we get

$$\bar{u}''(x''_{M'}, y''_{M'}) - \bar{i}\, u_x''(-x,y) + \bar{j}\, u_y''(- x, y) \tag{87}$$

and therefore

$$\bar{i}\left[u_x''(- x,y) - u_x(x,y)\right] + \bar{j}\left[u_y''(- x,y) + u_y(x,y)\right] = 0$$

Hence, we get the desired relations:

$$u_x''(-x,y) = u_x(x,y) \tag{88}$$

$$u_y''(-x,y) = - u_y(x,y) \tag{89}$$

The displacement vectors $\bar{u}'$ and $\bar{u}''$ at points P' and P'' are in reference to M_o' and M_o'' respectively. In order to obtain the corresponding displacement vectors with respect to M_o, we must introduce the relative displacement vectors of M_o' and M_o'' with respect to M_o, which are denoted by $\bar{U}_o, \bar{V}_o$. In general the relative displacement vectors for points along the line x = 0 and y = 0 are as follows:

$$\bar{U}(x,y) = \bar{i}\ U_x(y) + \bar{j}\ U_y(y) \tag{90}$$

$$\bar{V}(x,y) = \bar{i}\ V_x(x) + j\ V_y(x) \tag{91}$$

Because $U_y(x,y)$ = const and $V_x(x,y)$ = const and the symmetry relations are as follows, we have

$$\bar{U}(y) = -\ \bar{U}(x,-y)$$

and

$$\bar{V}(x) = -\ \bar{V}(-x)$$

Therefore,

$$U_y(y) = 0 \tag{92}$$

$$V_x(x) = 0 \tag{93}$$

and

$$U_x(y) = -\ U_x(-y) \tag{94}$$

$$V_y(x) = -\ V_y(-x) \tag{95}$$

Also

$$\bar{U}_o = \bar{i}\ U_x(0,b) \tag{96}$$

and

$$\bar{V}_o = \bar{j}\ V_y(2c,0) \tag{97}$$

The displacement vector at point P' relative to M_o is therefore

$$\bar{u}'_{TOT}(x,-y) = \bar{u}'(x,-y) + \bar{U}_o \tag{98}$$

and the displacement vector at point P" relative to M_o is similar; i.e.,

$$\bar{u}''_{TOT}(-x,y) = \bar{u}''(-x,y) + \bar{V}_o \tag{99}$$

With (77), (83), (96), and (97), we get

$$\bar{u}'_{TOT}(x,-y) = \bar{i}\left[u'_x(x,-y) + U_x(b)\right] + \bar{j}\, u'_y(x,-y) \tag{100}$$

$$\bar{u}''_{TOT}(-x,y) = \bar{i}\; u''_x(-x,y) + \bar{j}\left[u''_y(-x,y) + V_y(2c)\right] \tag{101}$$

These two relations are meaningful only when there are continuities of the displacement vector at the lines $y = \frac{b}{2}$ and $x = c$.

At these lines, we must have

$$\bar{u}'_{TOT}(x, +\frac{b}{2}) = \bar{u}(x, \frac{b}{2}) \tag{102}$$

and

$$\frac{\partial^n}{\partial y^n}\,\bar{u}''_{TOT}(x, -\frac{b}{2}) = \frac{\partial^n}{\partial x^n}\,\bar{u}(x, \frac{b}{2}) \tag{103}$$

$$(n = 1, 2, 3, \ldots\ldots)$$

also,

$$\bar{u}''_{TOT}(-c,y) = \bar{u}(c,y) \tag{104}$$

and

$$\frac{\partial^n}{\partial y^n}\,\bar{u}''_{TOT}(-c,y) = \frac{\partial^n}{\partial y^n}\,\bar{u}(c,y) \tag{105}$$

From (102), (100), and (72), we get

$$\bar{i}\left[u'_x(x,-\frac{b}{2}) - u(x,\frac{b}{2}) + U_x(b)\right] + j\left[u'_y(x,-\frac{b}{2}) - u_y(x,\frac{b}{2})\right] = 0 \tag{106}$$

Because of (81) and (82) and the fact that each component of the vector equation above must vanish, it follows that

$$u(x, \frac{b}{2}) = \frac{1}{2} U_x(b) \tag{107}$$

(107) represents one boundary condition along line y = b (in Figure 28 line $\overline{CD}$).

From (103) (n = 1), we have

$$\bar{i}\left[\frac{\partial u'_x}{\partial x}(x, -\frac{b}{2}) - \frac{\partial u_x}{\partial x}(x, \frac{b}{2})\right] + \bar{j}\left[\frac{\partial u'_y}{\partial x}(x, -\frac{b}{2}) - \frac{\partial u_y}{\partial x}(x, +\frac{b}{2})\right] = 0 \tag{108}$$

From (9), (10) and their partial differentiation with respect to x at y = $\frac{b}{2}$, we get

$$\frac{\partial u'_x}{\partial x}(x, -\frac{b}{2}) = -\frac{\partial u_x}{\partial x}(x, \frac{b}{2}) \tag{109}$$

and

$$\frac{\partial u'_y}{\partial x}(x, -\frac{b}{2}) = \frac{\partial u_y}{\partial x}(x, \frac{b}{2}) \tag{110}$$

Equations (108), (109), and (110) give

$$\frac{\partial u_x}{\partial x}(x, \frac{b}{2}) = 0 \tag{111}$$

Differentiation of (82) with respect to y at y = $\frac{b}{2}$ gives

$$-\frac{\partial u'_y}{\partial y}(x, -\frac{b}{2}) = \frac{\partial u_y}{\partial y}(x, \frac{b}{2}) \tag{112}$$

and with (82) follows

$$-\frac{\partial u_y}{\partial y}(x, \frac{b}{2}) = \frac{\partial u_y}{\partial y}(x, \frac{b}{2}) \tag{113}$$

or

$$\frac{\partial u_y}{\partial y}(x, \frac{b}{2}) = 0 \tag{114}$$

(111) and (114) issue the second boundary conditions along line y = b. For any two-dimensional problem, the normal stress is a homogeneous linear function of $\frac{\partial u_x}{\partial x}$ and $\frac{\partial u_y}{\partial y}$. It therefore follows that

$$\sigma_x(x, \frac{b}{2}) = 0 \tag{115}$$

$$\sigma_y(x, \frac{b}{2}) = 0 \tag{116}$$

where (116) represents the second boundary condition along line y = b.

Similar boundary conditions follow from (104) and (105). First we obtain from (104), (99), (87), (72), and (97)

$$\bar{i}\left[u''_x(-c,y) - u_x(c,y)\right] + \bar{j}\left[u''_y(- c,y) - u_y(c,y) + V_y(2c) = \right]0 \tag{117}$$

and because of (88) and (89),

$$u_y(c,y) = \tfrac{1}{2} V_y(2c) \tag{118}$$

Equation (118) is the first boundary condition at line x = c.

Setting n = 1 in equation (105) and combining with (99) and (87), we get

$$\bar{i}\left[\frac{\partial u''_x}{\partial y}(- c,y) - \frac{\partial u_x}{\partial y}(c,y)\right] + \bar{j}\left[\frac{\partial u''_y}{\partial y}(- c,y) - \frac{\partial u_y}{\partial y}(c,y)\right] = 0 \tag{119}$$

The partial differentiation of (89) with respect to y at x = c gives

$$\frac{\partial u''_y}{\partial y}(- c,y) = - \frac{\partial u_y}{\partial y}(c,y) \tag{120}$$

From the vector equation (119), we obtain

$$\frac{\partial u''_y}{\partial y}(- c,y) = \frac{\partial u_y}{\partial y}(c,y) \tag{121}$$

CONCLUSION

The problem of the elastic medium under transverse shear load can be completely attacked by solving a special problem of the basic representative element ABCD of Figure 27. This rectangle has been reproduced below for convenience.

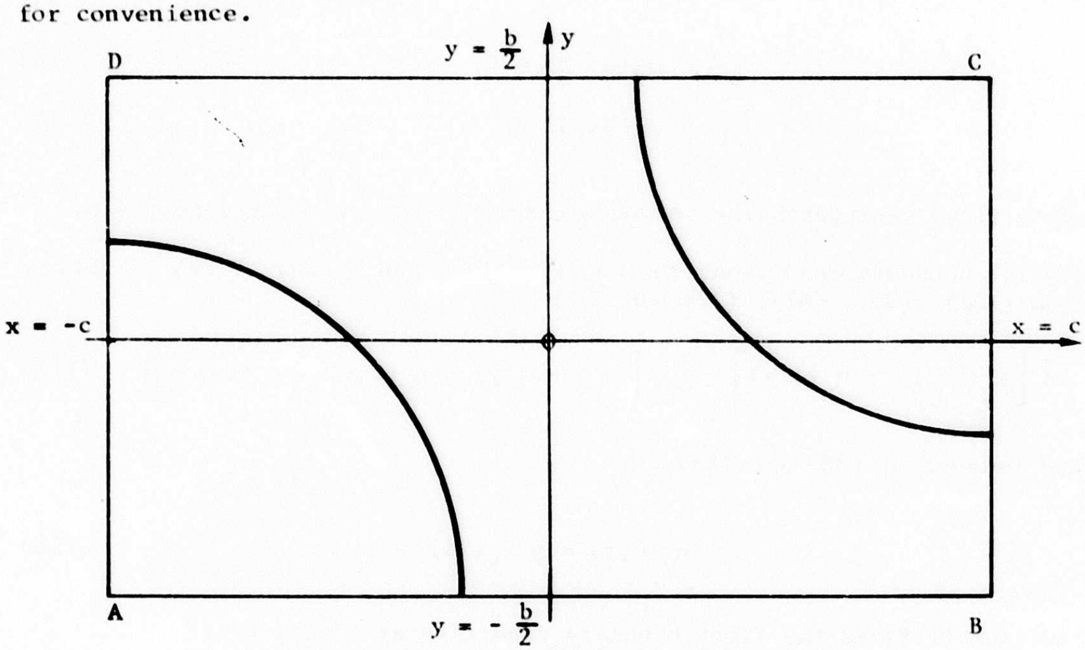


Figure 28. Rectangle ABCD.

From the analysis, the boundary conditions have been established for all boundary lines of Figure 28. For line CD, from equations (107) and (116), we have

$$u_x(x,\frac{b}{2}) = k_1 \qquad (129)$$

where k_1 is an orbitrary constant,

and

$$\sigma_y(x, \frac{b}{2}) = 0 \qquad (130)$$

$$-c \le x \le + c \qquad (131)$$

Then

$$\frac{\partial u_y}{\partial y}(c,y) = -\frac{\partial u_y}{\partial y}(c,y) \tag{122}$$

or

$$\frac{\partial u_y}{\partial y}(c,y) = 0 \tag{123}$$

Partial differentiation of (88) with respect to x at $x = c$ gives

$$-\frac{\partial u''_x}{\partial x}(-c,y) = \frac{\partial u_x}{\partial x}(c,y) \tag{124}$$

According to (88), the left-hand side function of (124) can be replaced; thereby

$$-\frac{\partial u_x}{\partial x}(c,y) = \frac{\partial u_x}{\partial x}(c,y) \tag{125}$$

is obtained. This means that

$$\frac{\partial u_x}{\partial x}(c,y) = 0 \tag{126}$$

For the same reason as pointed out above, and because of (123) and (126), it follows that

$$\sigma_x(c,y) = 0 \tag{127}$$

$$\sigma_y(c,y) = 0 \tag{128}$$

The first relation, equation (127), represents the second boundary condition for the line $x = c$ (in Figure 27 it is the line CB). (118) and (127) are the set of boundary conditions along line $x = c$.

Along line CB we obtain the two boundary conditions from (118) and (127)

$$u_y(c,y) = k_2 \tag{132}$$

where k_2 is another orbitrary constant,

and

$$\sigma_x(c,y) = 0 \tag{133}$$

for all values of y defined by the inequality

$$-\frac{b}{2} \le y \le \frac{b}{2} \tag{134}$$

In order to find the boundary conditions along the lines $\overline{AD}$ and $\overline{AB}$, we make use of the central symmetrical property of displacement with respect to the origin of Figure 28. Equation (74) expresses this property in analytic form and we obtain therefore from (129), (130), (132) and (133) with (74):

- for line AB,

$$u_x(-x, -\frac{b}{2}) = -k_1 \tag{135}$$

$$\sigma_y(-x, -\frac{b}{2}) = 0 \tag{136}$$

where

$$-c \le x \le +c \tag{137}$$

and for line $\overline{AD}$,

$$u_y(-c, -y) = -k_2 \tag{138}$$

$$\sigma_x(-c, -y) = 0 \tag{139}$$

where $$-\frac{b}{2} \le y \le \frac{b}{2} \tag{140}$$

With the boundary conditions (129) to (140), the problem of elasticity is defined and therefore unique solutions must exist. The boundary conditions are especially well suited for the method of finite elements and, therefore, directly applicable to the existing standard two-dimensional numerical program.

In our Fundamental Case T_{xy} of transverse shear, we take $k_1 = 0$ and $k_2 = 1$.

APPENDIX II

SYMMETRY RELATIONS IN PERFORATED PLATES

In the analysis of composite materials, it is necessary to consider a nonhomogenous medium consisting of a matrix and an array of fibers (or flakes). When the fibers are collimated, equally sized, and regularly arrayed, the symmetry of displacements in the composite can sometimes be deduced without recourse to complete solutions.

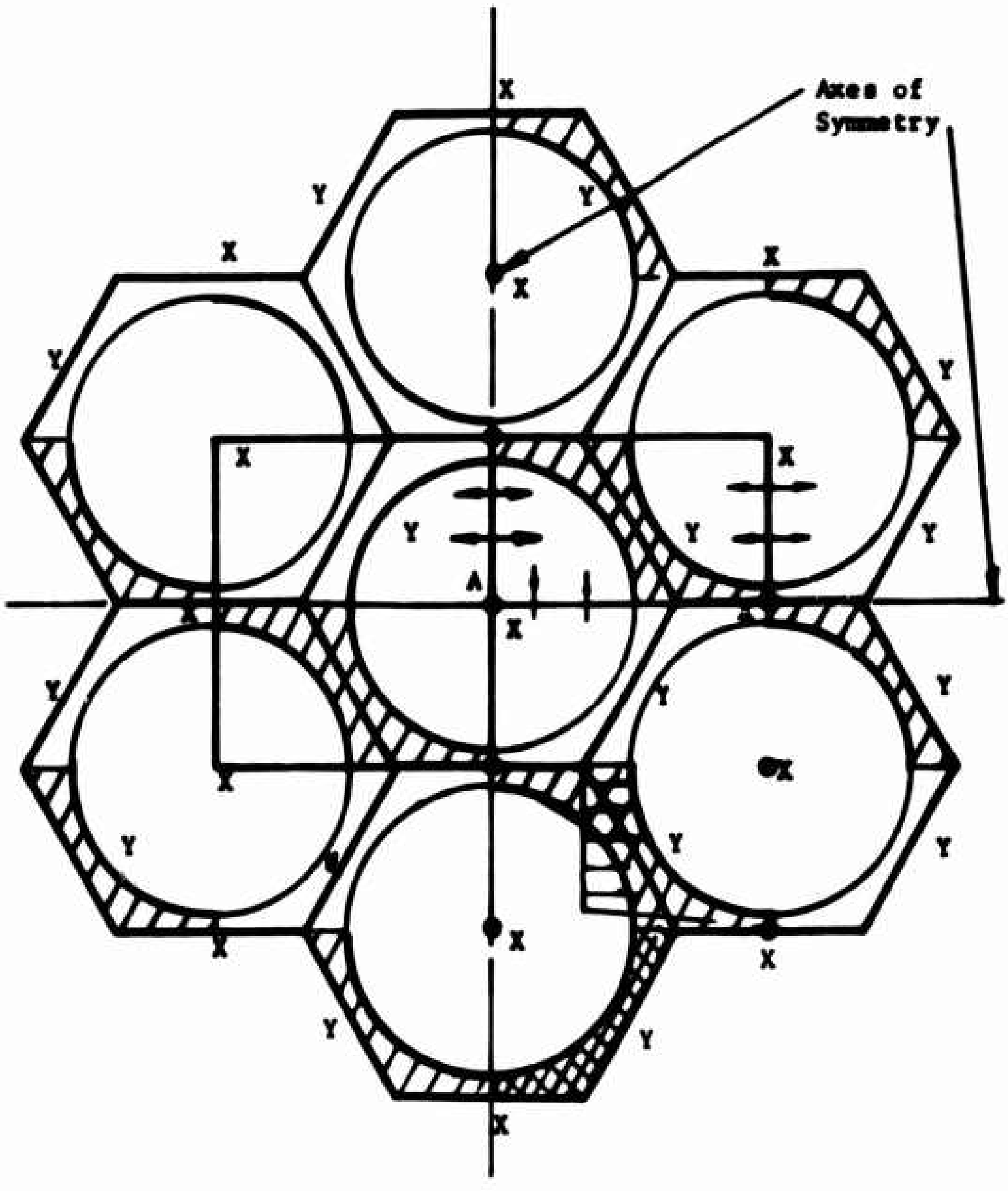


Figure 29. Basic Pattern for Fiber Spacing in Infinitely Large Cross Section (X, Y mark poles).

Let us consider a composite whose fibers are spaced in the hexagonal pattern shown in Figure 29. This pattern is repeated throughout the rectangular cross-section shown in Figure 30. The axes of geometry and physical symmetry pass through point A.

In many typical experiments performed on this model, the resultant loading is inevitably radially symmetric with respect to point A; that is, the applied force vector $\vec{F}(\vec{r})$ at any point $\vec{r}$ is the negative of the force vector at $-\vec{r}$ (see Figure 30).

$$\vec{F}(\vec{r}) = -\vec{F}(-\vec{r}) \tag{141}$$

In view of equation (141), the displacement response $\vec{u}(\vec{r},\vec{B})$ due to radially symmetric load system B is also radially symmetric; that is,

$$\vec{u}(\vec{r},B) = -\vec{u}(-\vec{r},B) \tag{142}$$

In Figure 30, therefore, only elements I and II need be considered further. Responses in elements III and IV can be described using equation (142).

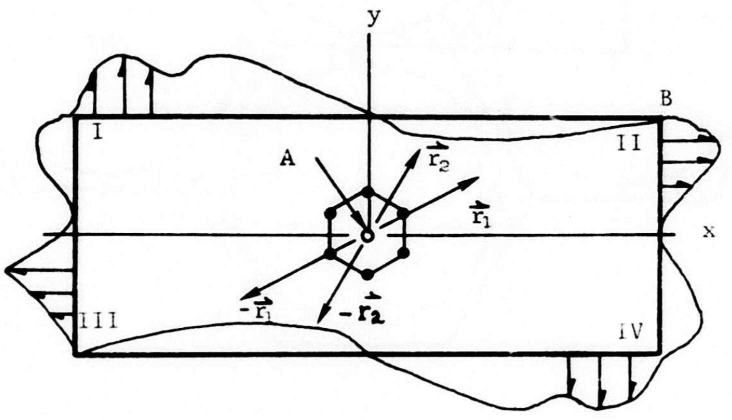


Figure 30. Cross Section Under Radially Symmetric Loading.

Another useful relation is obtained if the observer moves "behind" the section of Figure 30. Equivalently, the plate with its attached loads could be rotated 180° about the y-axis.* Since the ordinate is an axis of symmetry, the response of the rotated plate is the same as that of the original plate under a rotated load system B. See Figure 31 (a) and (b).

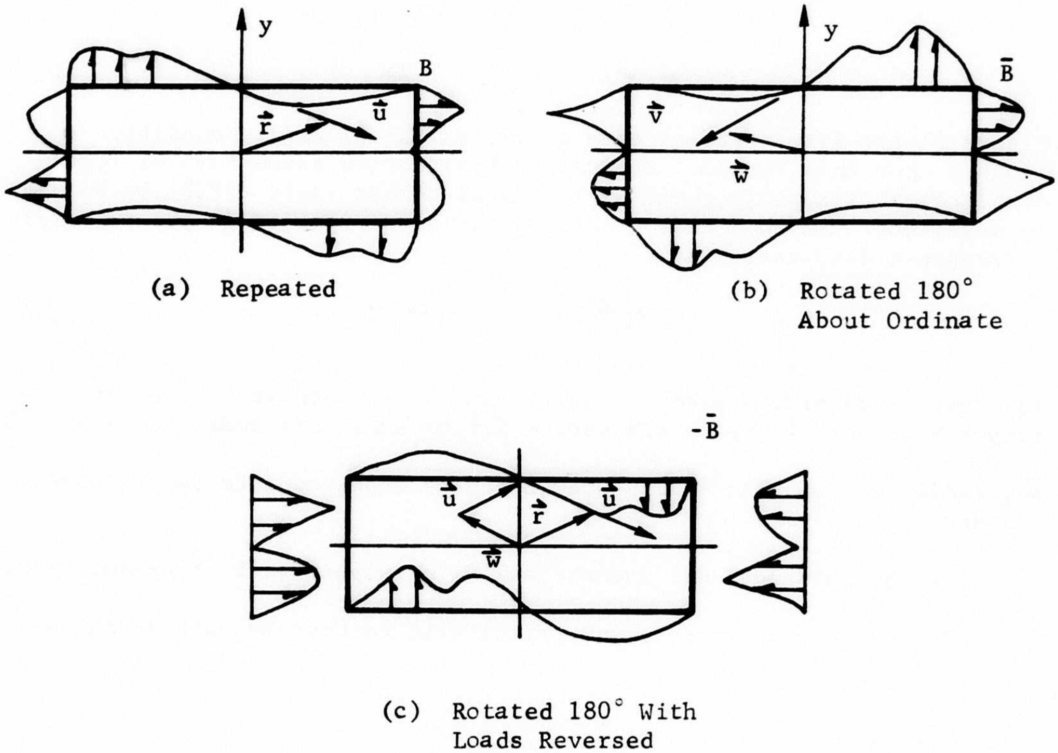


Figure 31. Illustration of Rotated System.

---

* Due to the symmetries, rotation about the x-axis would yield the same result.

This can be expressed symbolically if we define the vector operator $Rot_y[\vec{r}]$ as the mirror-imaging process just described. In this case, referring to Figure 31,

$$\vec{w} = Rot_y [\vec{r}] \tag{143}$$

$$\vec{v}(\vec{w},\overline{B}) = Rot_y [\vec{u}(\vec{r},B)] \tag{144}$$

$$\overline{B}(\vec{w}) = Rot_y \; B(\vec{r}) \tag{145}$$

Figure 31 (c) illustrates a consequence of the linearity condition imposed upon this system. The rather restrictive assumption of force-displacement linearity requires the displacement field $\vec{u}(\vec{r},\vec{B})$ to be small in magnitude. In this case, reversal of forces results in the reversal of response displacements.

$$\vec{v}(\vec{w},\overline{B}) = -\vec{v}(\vec{w},-\overline{B}) \tag{146}$$

In order to determine symmetry relations, it is necessary to relate the response vector $\vec{v}(\vec{w},B)$ to the vector $\vec{u}(\vec{r},B)$ under the same load system B.

Any radially symmetric load system B can be resolved into two component systems:

1. B_s, having loads symmetrical with respect to both x- and y-axes
2. B_a, having loads asymmetrical with respect to both x- and y-axes

These systems are illustrated in Figure 32.

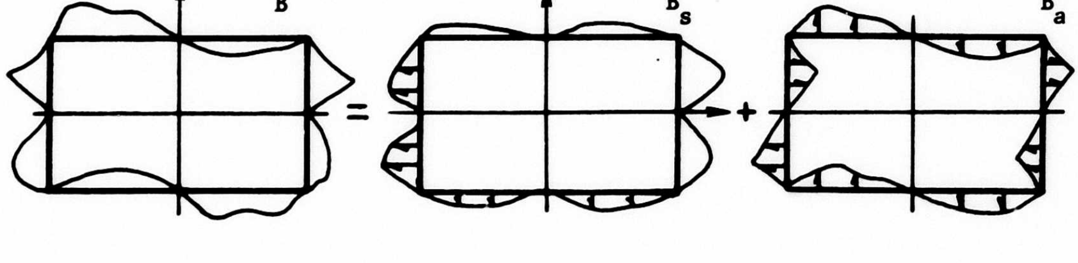


Figure 32. Resolution of Load System B.

If the rotation operations are performed on each component system, we find that symmetry or asymmetry of loading can be defined by equations (147) and (148).

$$\text{Symmetry:} \quad \bar{B}_s(\overleftrightarrow{w}) = \text{Rot}_y\left[B_s(\overleftrightarrow{r})\right] = B_s(\overleftrightarrow{w}) \tag{147}$$

$$\text{Asymmetry:} \quad \bar{B}_a(\overleftrightarrow{w}) = \text{Rot}_y\left[B_a(\overleftrightarrow{r})\right] = -\left[B_a(\overleftrightarrow{w})\right] \tag{148}$$

Upon application to equation (144), the following relationships are obtained

$$\vec{v}(\vec{w},\bar{B}_s) = \vec{v}(\vec{w},B_s) = \text{Rot}_y\left[\vec{u}(\vec{r},B_s)\right] \tag{149}$$

That is,

$$\vec{v}(\vec{w},B_s) = \text{Rot}_y\left[\vec{u}(\vec{r},B_s)\right] \tag{150}$$

$$\vec{v}(\vec{w},\bar{B}_a) = \vec{v}(\vec{w},-B_a) = -\vec{v}(\vec{w},B_a) = \text{Rot}_y\left[\vec{u}(\vec{r},B_a)\right] \tag{151}$$

and

$$\vec{v}(\vec{w},B_a) = -\text{Rot}_y\left[\vec{u}(\vec{r},B_a)\right] \tag{152}$$

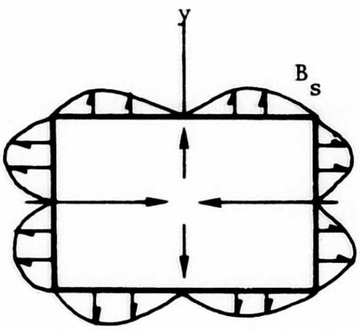
The response $\vec{v}$ at point $\vec{w}$ in element I is determined from the response $\vec{u}$ at $\vec{r}$ in element II by rotating $\vec{u}(\vec{r})$ and then multiplying by ± 1. It is important to note that only <u>one</u> element need by considered under either load system.

Some boundary conditions can be obtained by considering conditions along the y axis. In this case, equations (150) and (152) yield

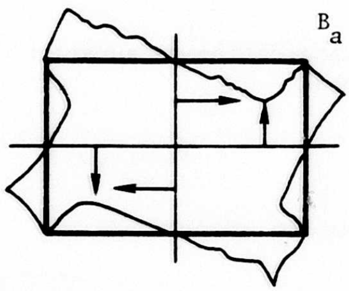
$$\vec{r}_y = \vec{w}_y, \ \vec{u}(\vec{r}_y,B) = \vec{v}(\vec{r}_y,B) \tag{153}$$

$$\begin{aligned} \vec{v}(\vec{r}_y,B_s) &= \text{Rot}_y\left[\vec{u}(\vec{r}_y,B_s)\right] = \vec{u}(\vec{r}_y,B_s) \\ \vec{v}(\vec{r}_y,B_a) &= -\text{Rot}_y\left[\vec{u}(\vec{r}_y,B_a)\right] = \vec{u}(\vec{r}_y,B_a) \end{aligned} \tag{154}$$

Equation (153) implies that $\vec{u}(\vec{r}_y, B_s)$ is parallel to the y-axis, while equation (154) implies that $\vec{u}(\vec{r}_y, B_a)$ is perpendicular to the y-axis. Typical boundary conditions are illustrated in Figure 33.



(a) Responses Along Axes due to Symmetric Load System B_s



(b) Responses Along Axes due to Asymmetric Load System B_a

Figure 33. Responses of Points on Axes of Symmetry.

DISCUSSION

The results of equations (149) through (154) hold true only if point A is (1) a pole of lead radial symmetry and (2) an origin of plate symmetry.

In an infinite plate under uniform shear loading at the boundaries, an infinite number of poles of type A exist. In Figure 29, these poles are indicated by X and Y. The coordinate axes (axes of symmetry) for points X are vertical and horizontal in this figure, while for points Y the coordinate systems are rotated. For the infinite plate only one sub-element, such as the one shaded in Figure 29, has to be analyzed to obtain a complete response description. For the finite plate, on the other hand, an entire quadrant must be analyzed. In a practical experiment, such a point A must exist, and the symmetries of the loading must be known if adequate verification is to be performed.

APPENDIX III

COMPUTER PROGRAM

INPUT DECK SETUP

In this program the six fundamental load cases are referred to as cases 1 through 6 as indicated in the following table:

	N_x	N_y	N_z	T_{yz}	T_{zx}	T_{xy}
case	1	2	3	4	5	6

FUNDAMENTAL CASES ONE THROUGH FOUR

CARD	COLUMNS	FORMAT	CONTAINING
1.	1-80	10A8	Run Identification
2.	1-10	E10.3	V_f
	11-20	E10.3	E_I/E_{II}
	21-30	E10.3	E_I
	31-40	E10.3	v_I
	41-50	E10.3	v_I/v_{II}

Repeat cards 1 and 2 up to 10 runs

3. Two blank cards end data

FUNDAMENTAL CASES FOUR AND FIVE

CARD	COLUMNS	FORMAT	CONTAINING
1.	1-80	10A8	Run Identification
2.	1-10	E10.3	G_I
	11-20	E10.3	G_{II}
	21-30	E10.3	V_f

Repeat cards 1a and 2 for same runs made by cases one through four

3. Two blank cards end data

PROGRAM FENG

CARD	COLUMNS	FORMAT	CONTAINING
1.	1	I1	1 for 1st 6 files 2 for 2nd 6 files
2.	1-10	E10.3	V_f
	11-20	E10.3	E_I/E_{II}
	21-30	E10.3	E_I
	31-40	E10.3	v_I
	41-50	E10.3	v_I/v_{II}

Repeat card 2 NS times, in the order the runs were made in Phase I

3. One blank card ends the run

Because of core size limitations, a limit of 10 runs of six fundamental cases can be run at one time. Also, because only 20 cases were needed for this study, Program FENG was written to accept only two sets of six files, these to be run separately. If more files are to be desired on one tape, appropriate tape spacing logic must be added to the program.

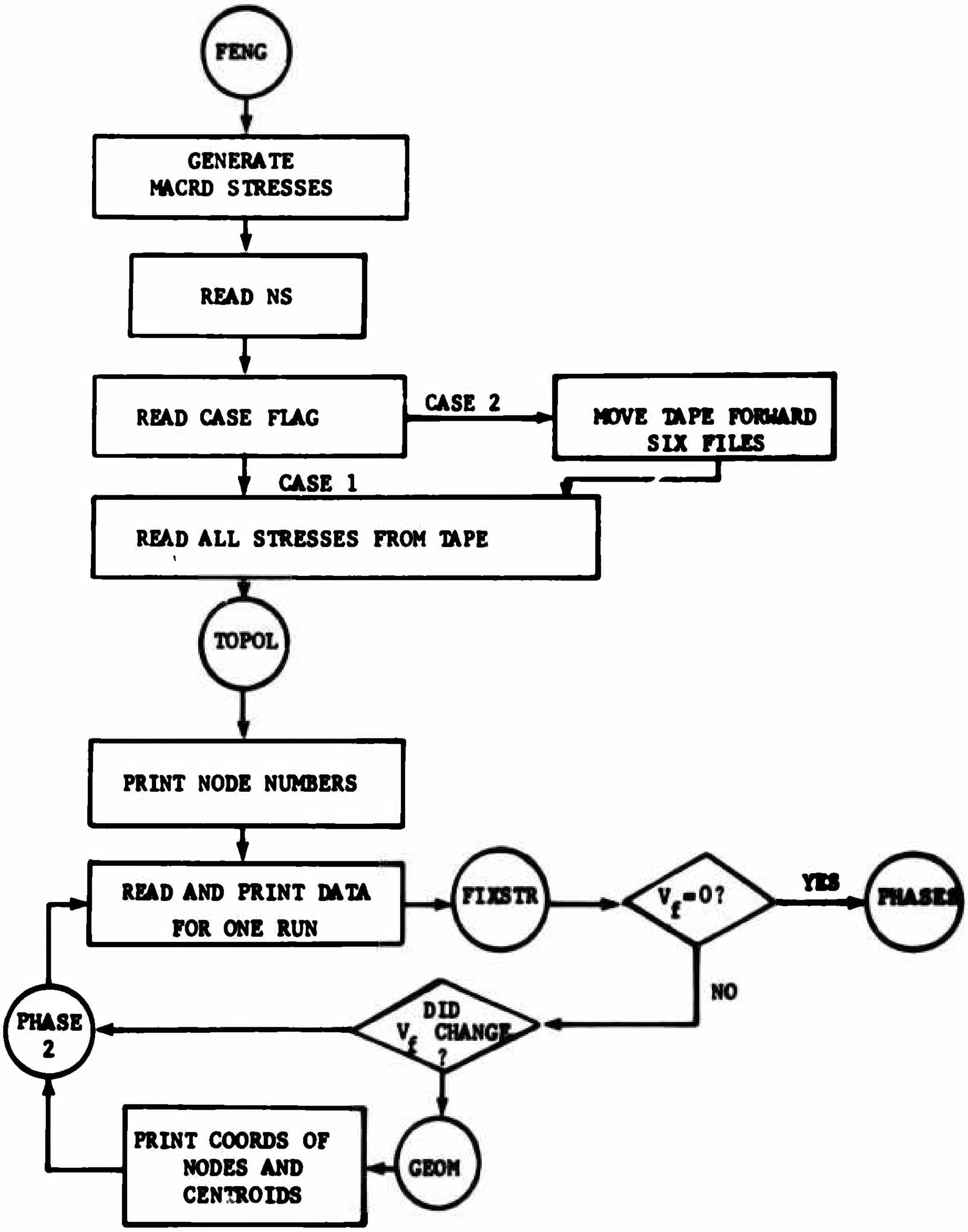
PROCEDURE FOR WRITING DATA TAPE 20

Before running Program FENG, it is necessary to write a data tape containing stress values for each of 79 triangular segments. This is accomplished by running sequentially Fundamental Case 1 through Fundamental Case 6, one or more times. Each fundamental case generates one file on Tape 20 which is an input to Program FENG.

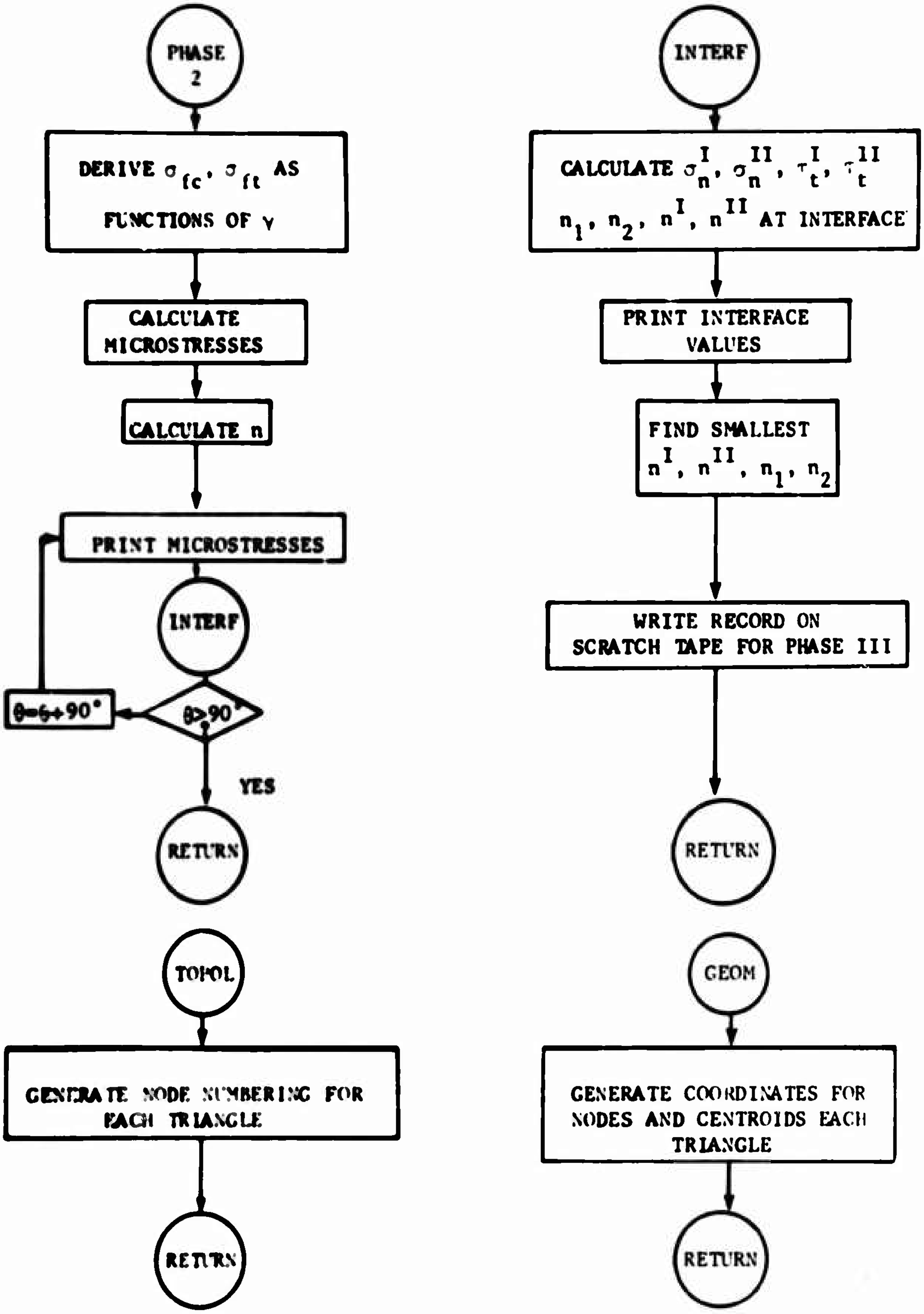
For each set of six files, the Program MAIN must be adjusted for cases 1 through 4, and Program LONTUD must be modified for cases 5 and 6, as follows:

- The first record of the first file on the tape must contain the integer N_s = number of runs per file.

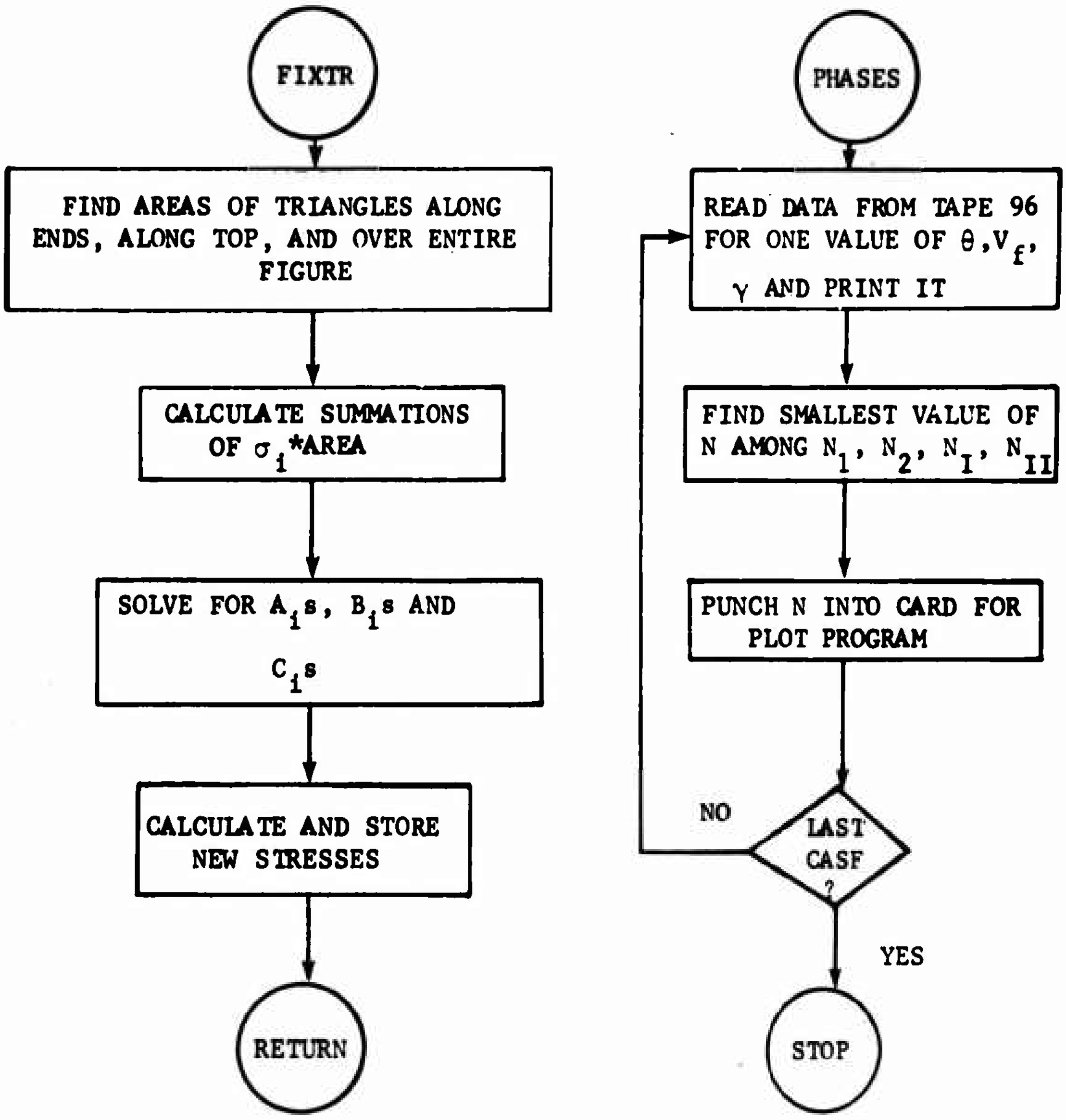
- Each Fundamental Case must space Tape 20 an appropriate number of files so as to leave room for files already written.



PROGRAM FENG



SUBROUTINES PHASE 2, TOPOL, GEOM, INTERF



SUBROUTINES FIXSTR (KASE), PHASES

PHASE ONE

FUNDAMENTAL CASES ONE, TWO, AND SIX
(TRANSVERSE LOADING)

FUNDAMENTAL CASE 1

```
      PROGRAM MAIN
C
C  THIS IS THE EXECUTIVE PROGRAM USED WITH SUBROUTINE BIGMX AND SUBROUTINE
C  CHLSKY  TO GENERATE AND SOLVE LARGE SYSTEMS OF LINEAR EQUATIONS
C                                                                       CASE 1
      REWIND 20
      CALL TAPESKIP(20,6,0)
    1 CONTINUE
      CALL INPUT
C
C  NOW HAVE ALL K-PRIME AND CDA MATRICES
C
      CALL CHLSKY
C
C  NOW HAVE SOLUTION U IN SP
C
      CALL STRESS
C
C  ALL STRESSES NOW PRINTED OUT
C
      GO TO 1
      END
```

```
      SUBROUTINE INPUT
C
C     TRANSVERSE NORMAL  CASE 1
C
C
C   THIS SUBROUTINE READS AND PRINTS THE INPUTS FOR THE PLANAR FINITE
C  ELEMENT PROGRAM.  ALL INPUTS NOT READ ARE GENERATED HERE.
C
      COMMON /STRSS/ ALF(155), MSIZE, R, GAMMA, EI, GII,XNU(2)
      COMMON /1/  S(32,32,5),L(300,3),G(32),SPACE(20)
     1          , NUF(3,314)         ,X(155),Y(155),CDA(3,6,250)
     2         , KPR(6,6,250),MSK(310),JTOTAL,N,KREM,M , IMR, IPL
     3          ,  P(155,2)   ,D(155,2)
      COMMON/LIM/  LIM1(10),LIM2(10)
      DIMENSION    SP(32,10)
      DIMENSION NUYX(250),NUXY(250),   GXY(250), EX(250), EY(250)
      DIMENSION BUMP(10) , BUMP1(5)
      EQUIVALENCE   (NUF,SP)
      EQUIVALENCE   (C,CP)
      DIMENSION JL(79,3)
      DIMENSION COMENT(10)
C
C  NUYX = NUXY, THIS MODIFICATION
      EQUIVALENCE   (NUXY,NUYX)
      DIMENSION   TH(155) , CP(3,3) , T1(3,3,2),T2(3,3),DTD(18),
     1  T(6,6),KMX(6,6), C(3,3),A(6,6,2),DO(3,6),DD(18),DTO(6,3)
      EQUIVALENCE   (S,F) , (S(901),VF), (S(1801),PHI),(S(2701),ER)
     1          , (S(3001),EF),(S(3901),NUR)
     2          ,  (DO,DD) , (DTO,DTD)   , (S(4356),TH)
     3       ,  (S(4511),CP) , (S(4521),T1), (S(4541),T2)
     4       ,  (S(4551),CABC) , (S(4581),T) , (S(4621),KMX)
     5    ,    (S(4671),A)
      TYPE REAL NUXY
      TYPE INTEGER  BUMP, BUMP1
      TYPE REAL     NUYX, NUR, NUF, KPR, KMX
      DATA  (KTOTAL = 97)
      DATA(BUMP=  2,4,7,7,15,1 ,9 ,9 ,12,-1)
      DATA(BUMP1= 5,7,7,7,5 )   ,(PI= 3.1415927)
      DATA  (RAD = 57.29578)
      DATA  (XLIM=1.E-8)
      DATA (PK=999.)
      DATA ((DD(JJ),JJ=1,18) = 0., 0., 0., 1., 0., 0., 0., 0., 1.,
     1                         0., 0., 0., 0., 0., 1., 0., 1., 0. )
      DATA ((DTD(JJ),JJ=1,18) = 0., 1., 0., 0., 0., 0., 0., 0., 0.,
     1                          0., 0., 1., 0., 0., 1., 0., 1., 0.)
      DATA(((JL(I,J),I=1,79),J=1,3) =
     1   1,1,1,2,2,3,3,3,4,4,4,5,6,7,7,8,8,8,9,10,10,11,11,12,13,13,
     2   14,15,15,16,16,17,17,18,18,19,20,23,24,24,25,25,25,26,28,29,
     3   29,30,30,35,21,21,21,21,22,22,23,23,27,27,27,28,28,36,46,37,
     4   47,38,48,39,39,40,41,41,41,42,42,43,43,
     5   2,3,4,6,7,7,8,9,9,10,11,11,13,6,14,14,15,16,16,16,17,10,
     6   18,18,20,21,21,21,22,15,23,23,24,17,25,25,35,27,27,28,28,29,
     7   30,30,32,32,33,33,34,36,35,37,38,39,39,40,40,41,41,42,43,43,44,
     8   45,37,46,38,47,39,48,49,49,40,50,51,51,52,52,53,
     9   3,4,5,7,3,8,9,4,10,11,5,12,14,14,8,15,16,9,10,17,18,18,12,
     1   19,21,14,15,22,23,23,17,24,25,25,19,26,21,24,28,25,29,30,26,
     2   31,29,33,30,34,31,37,37,38,39,22,40,23,41,27,42,43,28,44,32,
     3   46,36,47,37,48,38,49,40,50,50,51,42,52,43,53,44 )
C
      READ 1000,(COMENT(I),I=1,10)
 1000 FORMAT(10A8)
      PRINT 1001,(COMENT(I),I=1,10)
```

```
 1001 FORMAT(1H1,10A8)
C
C  KTOTAL - TOTAL NUMBER OF NODES CONSIDERED
C
C
C  READ AND PRINT
C
      JTOTAL = 158
      READ 1003, VF, GAMMA,EI, XNUI, BETA
 1003 FORMAT(5E10.4)
      IF (VF.EQ.0.)  GO TO 950
      PRINT 1018
      PRINT 1016
      PRINT 1017,  VF, GAMMA, EI, XNUI, BETA
 1016 FORMAT(1H ,13X,2HVF, 10X,5HGAMMA, 13X,2HEI, 11X,4HXNUI, 11X4HBETA)
 1017 FORMAT(1H ,5(E15.8))
 1018 FORMAT(////)

      S3 = SQRT(3.)
      S302 = S3/2.
      EII = EI / GAMMA
      XNUII = XNUI/BETA
      GII = EII/(2.*(1.+XNUII))
      XNU(1) = XNUI
      XNU(2) = XNUII
C
      EPI = EI
      EPII = EII
      XNUPI = XNUI
      XNUPII = XNUII
      EPI = EI/(1.-XNUI**2)
      EPII = EII/(1.-XNUII**2)
C
      XNUPI = XNUI/(1.-XNUI)
      XNUPII = XNUII/(1.-XNUII)
      DO 200 I = 1,97
      TH(I) =1.
      ALF(I)=0.
      D(I,1) = 1000.
      D(I,2) = 1000.
      P(I,1) = 0.
      P(I,2) = 0.
  200 CONTINUE
      I = 0
      DO 201  J= 1,10
      I = I +BUMP(J)
      P(I,2) = 1000.
      P(98-I,2) = 1000.
      D(I,2) = 0.
      D(98-I,2) = 0.
  201 CONTINUE
      I = 0
      DO  202 J = 1,5
      I = I+ BUMP1(J)
      P(I,1) = 1000.
      P(98-I,1) = 1000.
      D(I,1) = 1.
      D(98-I,1) = -1.
  202 CONTINUE
      P(1,1) =  1000.
      P(1,2) =  1000.
      D(1,1) =  1.
```

```
      D(1,2) =  0.
      P(34,1) = 1000.
      P(34,2) = 1000.
      D(34,1) = 1.
      D(34,2) = 0.
      P(64,1) = 1000.
      P(64,2) = 1000.
      D(64,1) = -1.
      D(64,2) =  0.
      P(97,1) = 1000.
      P(97,2) = 1000.
      D(97,1) = -1.
      D(97,2) =  0.
      R= SQRT(2.*S3*VF/PI)
      X(1) = S302
      Y(1) = .5
      DO 210 I=1,4
      X(I+1) = S302 - R/4.*COS(PI*(I-1)/6.)
      Y(I+1) = .5 -  R/4. *SIN(PI*(I-1)/6.)
  210 CONTINUE
C
      DO 220  I=1,7
      X(I+5) = S302 - R/2.* COS(PI*(I-1)/12.)
      Y(I+5) = .5    - R/2.* SIN(PI*(I-1)/12.)
C
      X(I+12) =  S302 - 3.*R/4.*COS(PI*(I-1)/12.)
      Y(I+12) =  .5  -  3.*R/4.*SIN(PI*(I-1)/12.)
C
      X(I+19) =  S302 - R * COS(PI*(I-1)/12.)
      Y(I+19) =  .5  -  R * SIN(PI*(I-1)/12.)
  220 CONTINUE
      X(34) = S302
      Y(34) = - .5
      X(31) = S302
      Y(31) = (Y(26)+Y(34)) / 2.
      X(45) = -.5* TAN(PI/6.)
      Y(45) = .5
      DX = (1.-R)/(2.*COS(PI/6.))
      X(36) = X(45) + DX
      Y(36) = .5
      X(35) = (X(20)+X(36))/2.
      Y(35) = .5
C
      DO 230 I = 1,4
      X(I+45) = (4 -I)* X(45) /4.
      Y(I+45) = (4 -I)* Y(45) /4.
  230 CONTINUE
C
      DELX = X(46) -X(45)
      DELY = Y(46)-Y(45)
      DO 240 I = 1,8
      X(I+36) = X(I+35) +DELX
      Y(I+36) = Y(I+35) +DELY
  240 CONTINUE
      X(32) = X(44) +(X(34)-X(44))/3.
      Y(32) = -.5
      X(33) = 2.*X(32) - X(44)
      Y(33) = -.5
      X(27) = X(23) + (X(32)-X(23))/3.
      Y(27) = Y(23) + (Y(32)-Y(23))/3.
      X(28) = 2.*X(27)-X(23)
      Y(28) = 2.*Y(27)-Y(23)
```

```
      X(29) = X(28) +(X(31)-X(28))/3.
      Y(29) = Y(28) +(Y(31)-Y(28))/3.
      X(30) = 2.*X(29)-X(28)
      Y(30) = 2.*Y(29)-Y(28)
C
C
      DO 250 I=50,97
      X(I) = -X(98-I)
      Y(I) = -Y(98-I)
  250 CONTINUE
C
C
C  PRINT OUT NODAL DATA
      I = 1
   12 CONTINUE
      LINE = 4
      PRINT 1010
   13 CONTINUE
      PRINT 1011, I, X(I), Y(I), P(I,1), P(I,2), D(I,1), D(I,2),
     1            TH(I), ALF(I)
      I = I+1
      IF(I.GT.KTOTAL)  GO TO 14
      LINE =  LINE + 2
      IF(LINE.GT.56) GO TO 12
      GO TO 13
   14 CONTINUE
 1011 FORMAT(1H  I4,7(3XE11.4),5X,F11.2/)
 1010 FORMAT(5H1NODE,7X,1HX,13X,1HY,13X,2HP1,12X,2HP2,12X,2HD1,12X,2HD2,
     1         8X,9HTHICKNESS,9X,5HALPHA /)
C
C  I       - NUMBER OF NODE
C  X       - X- COORDINATE OF ITH NODE
C  Y       - Y- COORDINATE OF ITH NODE
C  P(I,1)  - KNOWN AND UNKNOWN FORCE COMPONENTS ALONG 1 DIRECTION
C  P(I,2)  - KNOWN AND UNKNOWN FORCE COMPONENTS ALONG 2 DIRECTION
C  D(I,1)  - KNOWN AND UNKNOWN DISPLACEMENTS ALONG 1 DIRECTION
C  D(I,2)  - KNOWN AND UNKNOWN DISPLACEMENTS ALONG 2 DIRECTION
C  ALF(I)  - ANGLE BETWEEN X DIRECTION AND 1 DIRECTION (POSITIVE WHEN
C            COUNTER-CLOCKWISE)
C  TH(I)   - PLATE THICKNESS AT ITH NODE
C
C  GET MSK MATRIX
      JP = 0
      DO 27  J=1,KTOTAL
      DO 27  I=1, 2
      IF (P(J,I).GT.PK)  GO TO 27
      JP = JP+1
      MSK(JP) = 2 * (J-1) + I
   27 CONTINUE
C   MSK  IS MATRIX OF INDICES OF KNOWN FORCES
C   IF FORCE P IS UNKNOWN, IT IS INPUT AS 1000.
C  NOW READ IN TRIANGLE DATA
C
C  JTOTAL - TOTAL NUMBER OF TRIANGLES
      DO 19 I=1,79
      DO 19 J=1,3
   19 L(I,J) = JL(I,J)
      DO 260 I= 80,158
      DO 260 J= 1,3
      L(I,J) = 98 - L(159-I,J)
  260 CONTINUE
C  L(J,1) - INDEX OF THE FIRST  NODE OF THE JTH TRIANGLE
```

```
C  L(J,2) - INDEX OF THE SECOND NODE OF THE JTH TRIANGLE
C  L(J,3) - INDEX OF THE THIRD  NODE OF THE JTH TRIANGLE
C
      DO 20 I=1,JTOTAL
      IF(I.LE.36.OR.I.GE.123) GO TO 300
      EX(I) = EPII
      EY(I) = EPII
      NUXY(I) =XNUPII
      GXY(I) = EPII/(2.*(1.+XNUPII))
      GO TO 310
  300 EX(I) = EPI
      EY(I) = EPI
      NUXY(I) = XNUPI
      GXY(I) = EPI  / (2.*(1.+XNUPI))
  310 CONTINUE
   20 CONTINUE
C  PRINT OUT TRIANGLE DATA
      LINE = 4
      PRINT  1012
      DO 24 I=1,JTOTAL
      IF(LINE.LT.54)  GO TO 22
      LINE = 4
      PRINT 1012
 1012 FORMAT(1H1,8HTRIANGLE,4X6HNODE 1,3X,6HNODE 2,3X,6HNODE 3,18X,2HEX,
     1    18X 2HEY,16X,4HNUYX,17X,3HGXY//)
   22 LINE = LINE+2
   24 PRINT 1023, I, (L(I,J),J=1,3),EX(I),EY(I),NUYX(I),GXY(I)
 1023 FORMAT(1H ,5X,I3,7X,I3,2(6X,I3),4(5X,E15.8)/)
C
C
      MSIZE = 156
      KREM = 6
      N = 25
      M = MSIZE / N
      IF(KREM.NE.N)  M = M+1
C
      LIM1(1) = 1
      LIM1(2) = 1
      LIM1(3) = 1
      LIM1(4) = 1
      LIM1(5) = 1
      LIM1(6) = 1
      LIM1(7) = 1
      LIM2(1) = 158
      LIM2(2) = 158
      LIM2(3) = 158
      LIM2(4) = 158
      LIM2(5) = 158
      LIM2(6) = 158
      LIM2(7) = 158

C
C  PRINT OUT PARTITION INFORMATION
      PRINT 1008
 1008 FORMAT(1H1,12X,20HTRIANGLES CONSIDERED)
      DO 31  I=1,M
      ISIZE = N
      IF(I.EQ.1)  ISIZE = KREM
      PRINT 1009,  I,LIM1(I),LIM2(I),  ISIZE
   31 CONTINUE
 1009 FORMAT(10H PARTITION,6X 5HFIRST,2X,2HTO,2X,4HLAST, 6X,9HDIMENSION/
     1     4X,I3,11X,I3,6X,I3,10X,I3)
```

```
      DO 28 I = 1,KTOTAL
      ALF(I) = ALF(I)/RAD
   28 CONTINUE
C  ALL ANGLES NOW IN RADIANS
      DO 400 I=1, JTOTAL
C  I   IS A TRIANGLE COUNTER
      B11 = 1./EX(I)
      B12 = -NUYX(I) / EY(I)
      B22 = 1. / EY(I)
      B33 = 1. / GXY(I)
      DELTA = B11*B22 -B12**2
      CP(2,1)    = - B12 / DELTA
      CP(1,1)    = B22 / DELTA
      CP(3,1)    =  0.
      CP(1,2) = CP(2,1)
      CP(2,2)    = B11/ DELTA
      CP(3,2)    =  0.
      CP(1,3)    =  0.
      CP(2,3)    =  0.
      CP(3,3) = 1./B33
C
C  C NOW IN C(3,3)   , MATRIX I
C
   30 CONTINUE
      THOMEG   = X(L(I,2))*Y(L(I,3)) +X(L(I,1))*Y(L(I,2))
     1         + Y(L(I,1))*X(L(I,3)) -X(L(I,3))*Y(L(I,2))
     2         - X(L(I,1))*Y(L(I,3)) -Y(L(I,1))*X(L(I,2))
      THOMEG=(TH(L(I,1))+TH(L(I,2))+TH(L(I,3)))/ 6. * THOMEG
      XI2 = X(L(I,2))- X(L(I,1))
      XI3 = X(L(I,3))- X(L(I,1))
      ETA2= Y(L(I,2))- Y(L(I,1))
      ETA3= Y(L(I,3))- Y(L(I,1))
      DELTA = XI2*ETA3 - XI3*ETA2
      DO 38 II = 1,3
      DO 38 JJ = 1,3
      A(II+3,JJ,1) = 0.
      A(II,JJ+3,1) = 0.
      A(II+3,JJ,2) = 0.
      A(II,JJ+3,2) = 0.
   38 CONTINUE
      A(1,1,1) = 1.
      A(2,1,1) = -(ETA3-ETA2) / DELTA
      A(3,1,1) = (XI3 - XI2) / DELTA
      A(1,2,1) = 0.
      A(2,2,1) = ETA3 / DELTA
      A(3,2,1) = -XI3 / DELTA
      A(1,3,1) = 0.
      A(2,3,1) = -ETA2 / DELTA
      A(3,3,1) = XI2 / DELTA
      DO 39 II = 1,3
      DO 39 JJ = 1,3
      A(II+3,JJ+3,1) = A(II,JJ,1)
      A(II,JJ,2) = A(JJ,II,1)
      A(II+3,JJ+3,2) = A(JJ,II,1)
   39 CONTINUE
C
C  TRANSPOSE OF INVERSE OF A NOW IN A(1,1,2) , A INVERSE STILL IN A
C
      CALL MXMULT(DO,A,KMX(1,1),3,6,6)
      CALL MXMULT(C,KMX(1,1)   ,CDA(1,1,I),3,3,6)
C  PRODUCT C*D*(A**-1) NOW IN CDA(1,1,I), ITH TRIANGLE
C
```

```
      CALL MXMULT(A(1,1,2),DT(,A,6,6,3)
      CALL MXCON(A,KPR(1,1,I),THOMEG,6,3)
      CALL MXMULT(KPR(1,1,I),CDA(1,1,I),KMX,6,3,6)
C
C   MATRIX K(I) NOW IN KMX, TRIANGLE I
      DO 40 II =1,6
      DO 40 JJ =1,6
      T(II,JJ)    = 0.
   40 CONTINUE
      T(1,1)    =  COS(ALF(L(I,1)))
      T(4,1)    =  SIN(ALF(L(I,1)))
      T(2,2)    =  COS(ALF(L(I,2)))
      T(5,2)    =  SIN(ALF(L(I,2)))
      T(3,3)    =  COS(ALF(L(I,3)))
      T(6,3)    =  SIN(ALF(L(I,3)))
      T(1,4)    = -T(4,1)
      T(4,4)    =  T(1,1)
      T(2,5)    = -T(5,2)
      T(5,5)    =  T(2,2)
      T(3,6)    = -T(6,3)
      T(6,6)    =  T(3,3)
      CALL MXMULT (KMX,T,A,6,6,6)
      T(4,1) = - T(4,1)
      T(5,2) = - T(5,2)
      T(6,3) = - T(6,3)
      T(1,4) = - T(1,4)
      T(2,5) = - T(2,5)
      T(3,6) = - T(3,6)
C
C   INVERSE OF T NOW IN T
C
      CALL MXMULT (T,A,KPR(1,1,I),6,6,6)
C
C  K-PRIME NOW IN KPR , A  HAS BEEN CLOBBERED.
  400 CONTINUE
      RETURN
  800 PRINT 1051,II,JJ,II
 1051 FORMAT(1H1, 5H  EF(,I3,1H,I3,7H) = ER(,I3,6H) = 0.)
      STOP
  900 CONTINUE
      PRINT 1050    , J
 1050 FORMAT(1H1  35HCOULD NOT INVERT MATRIX   T1,TRIANGLE,I3)
  950 CONTINUE
      REWIND 20                                                   CASE 1
      STOP
      END
```

```
      SUBROUTINE STRESS
C
C     STRESS SUBROUTINE   CASE 1
C
C
C  THIS SUBROUTINE DERIVES AND PRINTS STRESSES
C
      COMMON /STRSS/ ALF(155), MSIZE, R, GAMMA, E1, G11,XNU(2)
      COMMON /1/  S(32,32,5),L(300,3),G(32),SPACE(20)
     1          , NUF(3,314)        ,X(155),Y(155),CDA(3,6,250)
     2          , KPR(6,6,250),MSK(310),JTOTAL,N,KREM,M , IMR, IPL
     3          ,  P(155,2)   ,D(155,2)
      COMMON/LIM/  LIM1(10),LIM2(10)
      DIMENSION  DVX(6), SIGOUT(632)
      DIMENSION   SP(32,10)
      EQUIVALENCE(S,SIGOUT) ,(NUF,SP)
      TYPE REAL KPR
      EQUIVALENCE  (S,SIG) , (S(931),PSTR) , (S(1861),X0),(S(2161),Y0)
      EQUIVALENCE  (S(2461),DEL) , (S(2761),DX)
      DIMENSION   ERR(310)
      DIMENSION   DX(6) , SIG(4,168)  ,   PSTR(3,310)
      DIMENSION  X0(300) , Y0(300), DEL(310)
      DIMENSION  KS22(32,96)
      EQUIVALENCE   (KPR,KS22)
      TYPE REAL KS22
      DATA (PK=999.)
C  REMOVE GAPS FROM SP(32,10) = DEL(310)
      DO 5 J=1,KREM
      DEL(J) = SP(J,1)
    5 CONTINUE
      KLOC = KREM -N
      DO 10 I = 2,M
      KLOC = KLOC + N
      DO 10 J = 1,N
      DEL(KLOC+J) = SP(J,I)
   10 CONTINUE
      PRINT 1010
 1010 FORMAT(1H1,50X,13HDISPLACEMENTS,//)
      NDEL = MSIZE/7
      JCNT = 0
      DO 15 J=1,NDEL
      JCNT = JCNT + 1
      IF(JCNT.LE.18)  GO TO 14
      PRINT 1010
      JCNT = 0
   14 JFIR = 7*(J-1) + 1
      JLAST =   JFIR + 6
      PRINT 1011 , (K,K=JFIR,JLAST)
 1011 FORMAT(1H ,7(8X,4HDEL(,I3,1H) ))
      PRINT 1012, (DEL(K),K=JFIR,JLAST)
 1012 FORMAT(1H ,7(2X,E14.7)/)
   15 CONTINUE
      LOC1 = 7*NDEL+1
      LOC2 = MSIZE
      IF(LOC1.GT.LOC2)  GO TO 20
      PRINT 1011,(K,K=LOC1,LOC2)
      PRINT 1012, (DEL(K),K=LOC1,LOC2)
   20 CONTINUE
      DO 855  I=1,JTOTAL
      J = I
      KZ = 0
      DO 831 KK =1,3
```

```
      DO 832 KJ =1,2
      KZ = KZ + 1
      IF( P(L(J,KK),KJ).GT.PK) GO TO 828
      IF(2*(L(J,KK)-1)+KJ-MSK(IPL)) 801, 802, 803
  802 II2 = IPL
      GO TO 827
  801 IMR = IPL - 1
  806 IF(2*(L(J,KK)-1) +KJ -MSK(IMR))  804, 805, 805
  805 II2 = IMR
      IPL = IMR
      GO TO 827
  804 IMR = IMR - 1
      GO TO 806
  803 IMR = IPL + 1
  807 IF(2*(L(J,KK)-1) +KJ -MSK(IMR)) 805, 805, 810
  810 IMR = IMR + 1
      GO TO 807
  827 DVX(KZ) = DEL(II2)
      GO TO 832
  828 DVX(KZ)= D(L(J,KK),KJ)
  832 CONTINUE
      DX(KZ-1) = DVX(KZ-1)*COS(ALF(L(J,KK)))-DVX(KZ)*SIN(ALF(L(J,KK)))
      DX(KZ) = DVX(KZ-1)*SIN(ALF(L(J,KK)))+DVX(KZ)*COS(ALF(L(J,KK)))
  831 CONTINUE
      DX2 = DX(2)
      DX(2) = DX(3)
      DX(3) = DX(5)
      DX(5) = DX(4)
      DX(4) = DX2
      CALL MXMULT(CDA(1,1,I),DX, SIG(1,I),3,6,1)
      KK3 = 1
      IF(I.GE.37.AND.I.LE.122)  KK3 = 2
      SIG(4,I) = SIG(3,I)
      SIG(3,I) = XNU(KK3)*(SIG(1,I)+SIG(2,I))
C   SIGMA NOW IN SIG(1,I)   , TRIANGLE I
      XO(I) = 0.
      YO(I) = 0.
      DO 840  K = 1,3
      XO(I) = XO(I) + X(L(I,K))
      YO(I) = YO(I) + Y(L(I,K))
  840 CONTINUE
      XO(I) = XO(I) / 3.
      YO(I) = YO(I) / 3.
  850 CONTINUE
  855 CONTINUE
      ENBARX = R/4.*(SIG(1,3)+SIG(1,12)+SIG(1,24)+SIG(1,36) )
     1                + (1.-R)/2. *(SIG(1,44)+SIG(1,49) )
      ENBARX = 1./ENBARX
      DO 860 I=1,632
  860 SIGOUT(I) = SIGOUT(I)*ENBARX
      I = 1
  870 CONTINUE
      PRINT 1000
      LINE = 3
 1000 FORMAT(1H1,2(8HTRIANGLE,6X,8HCENTROID,9X,12HVECTOR SIGMA,10X)/)
  871 CONTINUE
      II = I+1
      DO  880  J=1,4
      GO TO (873,874,875,875), J
  873 PRINT 1001, I, XO(I), SIG(J,I),II,XO(II), SIG(J,II)
      GO TO 878
  874 PRINT 1002,    YO(I),SIG(J,I),      YO(II), SIG(J,II)
```

```
      GO TO 878
  875 PRINT 1003, SIG(J,I), SIG(J,II)
  878 CONTINUE
  880 CONTINUE
      LINE = LINE + 6
      I = I+ 2
      PRINT 1004
 1001 FORMAT(1H 2(5X,I3,F14.6, 7X,E15.8,10X))
 1002 FORMAT(1H 2(8X,F14.6, 7X,E15.8,10X) )
 1003 FORMAT(1H 2(29X,E15.8,10X))
 1004 FORMAT(/)
 1069 FORMAT(1H ,30X,E15.8)
      IF(I.GT.JTOTAL)  GO TO 890
      IF (LINE.GT.54) GO TO 870
      GO TO 871
  890 CONTINUE
      WRITE (20)   (SIGOUT(I),I=1,316)                                    CASE 1
      PRINT 8787, ENBARX
 8787 FORMAT(//23H NORMALIZATION FACTOR =,2X,E15.7)
      RETURN
      END
```

2026 CARDS

```
      PROGRAM MAIN
C
C  THIS IS THE EXECUTIVE PROGRAM USED WITH SUBROUTINE BIGMX AND SUBROUTINE
C  CHLSKY  TO GENERATE AND SOLVE LARGE SYSTEMS OF LINEAR EQUATIONS
C
      REWIND 20                                                           CASE 2
      CALL TAPESKIP(20,6,0)
      CALL TAPESKIP(20,1,0)
    1 CONTINUE
      CALL INPUT
C
C  NOW HAVE ALL K-PRIME AND CDA MATRICES
C
      CALL CHLSKY
C
C  NOW HAVE SOLUTION U IN SP
C
      CALL STRESS
C
C  ALL STRESSES NOW PRINTED OUT
C
      GO TO 1
      END
```

23 CARDS

FUNDAMENTAL CASE 2

```
      SUBROUTINE INPUT
C
C     TRANSVERSE NORMAL- CASE 2
C
C
C   THIS SUBROUTINE READS AND PRINTS THE INPUTS FOR THE PLANAR FINITE
C  ELEMENT PROGRAM.  ALL INPUTS NOT READ ARE GENERATED HERE.
C
      COMMON /STRSS/ ALF(155), MSIZE, R, GAMMA, EI, GII,XNU(2)
      COMMON /1/  S(32,32,5),L(300,3),G(32),SPACE(20)
     1         , NUF(3,314)        ,X(155),Y(155),CDA(3,6,250)
     2         , KPR(6,6,250),MSK(310),JTOTAL,N,KREM,M , IMR, IPL
     3         ,  P(155,2)   ,D(155,2)
      COMMON/LIM/  LIM1(10),LIM2(10)
      DIMENSION   SP(32,10)
      DIMENSION NUYX(250),NUXY(250),  GXY(250), EX(250), EY(250)
      DIMENSION BUMP(10) , BUMP1(5)
      EQUIVALENCE  (NUF,SP)
      EQUIVALENCE  (C,CP)
      DIMENSION JL(79,3)
      DIMENSION COMENT(10)
C
C   NUYX = NUXY, THIS MODIFICATION
      EQUIVALENCE  (NUXY,NUYX)
      DIMENSION  TH(155) , CP(3,3) , T1(3,3,2),T2(3,3),DTD(18),
     1  T(6,6),KMX(6,6), C(3,3),A(6,6,2),D0(3,6),DD(18),DT0(6,3)
      EQUIVALENCE  (S,F) , (S(901),VF), (S(1801),PHI),(S(2701),ER)
     1           , (S(3001),EF),(S(3901),NUR)
     2           ,  (D0,DD) , (DT0,DTD)   , (S(4356),TH)
     3       ,  (S(4511),CP) , (S(4521),T1), (S(4541),T2)
     4       ,  (S(4551),CABC) , (S(4581),T) , (S(4621),KMX)
     5     ,   (S(4671),A)
      TYPE REAL NUXY
      TYPE INTEGER  BUMP, BUMP1
      TYPE REAL     NUYX, NUR, NUF, KPR, KMX
      DATA  (KTOTAL = 97)
      DATA(BUMP=  2,4,7,7,15,1 ,9 ,9 ,12,-1)
      DATA(BUMP1= 5,7,7,7,5 )  ,(PI= 3.1415927)
      DATA  (RAD = 57.29578)
      DATA  (XLIM=1.E-8)
      DATA (PK=999.)
      DATA ((DD(JJ),JJ=1,18) = 0., 0., 0., 1., 0., 0., 0., 0., 1.,
     1                         0., 0., 0., 0., 0., 1., 0., 1., 0. )
      DATA ((DTD(JJ),JJ=1,18) = 0., 1., 0., 0., 0., 0., 0., 0., 0.,
     1                          0., 0., 1., 0., 0., 1., 0., 1., 0.)
      DATA(((JL(I,J),I=1,79),J=1,3) =
     1   1,1,1,2,2,3,3,3,4,4,4,5,6,7,7,8,8,8,9,10,10,11,11,12,13,13,
     2   14,15,15,16,16,17,17,18,18,19,20,23,24,24,25,25,25,26,28,29,
     3   29,30,30,35,21,21,21,21,22,22,23,23,27,27,27,28,28,36,46,37,
     4   47,38,48,39,39,40,41,41,41,42,42,43,43,
     5   2,3,4,6,7,7,8,9,9,10,11,11,13,6,14,14,15,16,16,16,17,10,
     6   18,18,20,21,21,21,22,15,23,23,24,17,25,25,35,27,27,28,28,29,
     7   30,30,32,32,33,33,34,36,35,37,38,39,39,40,40,41,41,42,43,43,44,
     8   45,37,46,38,47,39,48,49,49,40,50,51,51,52,52,53,
     9   3,4,5,7,3,8,9,4,10,11,5,12,14,14,8,15,16,9,10,17,18,18,12,
     1   19,21,14,15,22,23,23,17,24,25,25,19,26,21,24,28,25,29,30,26,
     2   31,29,33,30,34,31,37,37,38,39,22,40,23,41,27,42,43,28,44,32,
     3   46,36,47,37,48,38,49,40,50,50,51,42,52,43,53,44 )
C
      READ 1000,(COMENT(I),I=1,10)
 1000 FORMAT(10A8)
      PRINT 1001,(COMENT(I),I=1,10)
```

```
 1001 FORMAT(1H1,10A8)
C
C  KTOTAL - TOTAL NUMBER OF NODES CONSIDFRED
C
C
C  READ AND PRINT
C
      JTOTAL = 158
      READ 1003, VF, GAMMA,EI, XNUI, BETA
 1003 FORMAT(5E10.4)
      IF (VF.EQ.0.)  GO TO 950
      PRINT 1018
      PRINT 1016
      PRINT 1017,  VF, GAMMA, EI, XNUI, BETA
 1016 FORMAT(1H ,13X,2HVF, 10X,5HGAMMA, 13X,2HEI, 11X,4HXNUI, 11X4HBETA)
 1017 FORMAT(1H ,5(E15.8))
 1018 FORMAT(////)

      S3 = SQRT(3.)
      S302 = S3/2.
      EII = EI / GAMMA
      XNUII = XNUI/BETA
      GII = EII/(2.*(1.+XNUII))
      XNU(1) = XNUI
      XNU(2) = XNUII
C
      EPI = EI
      EPII = EII
      XNUPI = XNUI
      XNUPII = XNUII
      EPI = EI/(1.-XNUI**2)
      EPII = EII/(1.-XNUII**2)
C
      XNUPI = XNUI/(1.-XNUI)
      XNUPII = XNUII/(1.-XNUII)
      DO 200 I = 1,97
      TH(I) =1.
      ALF(I)=0.
      D(I,1) = 1000.
      D(I,2) = 1000.
      P(I,1) = 0.
      P(I,2) = 0.
  200 CONTINUE
      I = 0
      DO 201  J= 1,10
      I = I +BUMP(J)
      P(I,2) = 1000.
      P(98-I,2) = 1000.
      D(I,2) = 1.
      D(98-I,2) = -1.
  201 CONTINUE
      I = 0
      DO  202 J = 1,5
      I = I+ BUMP1(J)
      P(I,1) = 1000.
      P(98-I,1) = 1000.
      D(I,1) = 0.
      D(98-I,1) = 0.
  202 CONTINUE
      P(1,1) = 1000.
      P(1,2) = 1000.
      D(1,1) =  0.
```

```
      D(1,2) =  1.
      P(34,1) = 1000.
      P(34,2) = 1000.
      D(34,1) = 0.
      D(34,2) = -1.
      P(64,1) = 1000.
      P(64,2) = 1000.
      D(64,1) = 0.
      D(64,2) =  1.
      P(97,1) = 1000.
      P(97,2) = 1000.
      D(97,1) =  0.
      D(97,2) =  -1.
      R= SQRT(2.*S3*VF/PI)
      X(1) = S302
      Y(1) = .5
      DO 210 I=1,4
      X(I+1) = S302 - R/4.*COS(PI*(I-1)/6.)
      Y(I+1) = .5 -  R/4. *SIN(PI*(I-1)/6.)
  210 CONTINUE
C
      DO 220  I=1,7
      X(I+5) = S302 - R/2.* COS(PI*(I-1)/12.)
      Y(I+5) = .5   - R/2.* SIN(PI*(I-1)/12.)
C
      X(I+12) =  S302 - 3.*R/4.*COS(PI*(I-1)/12.)
      Y(I+12) =  .5  -  3.*R/4.*SIN(PI*(I-1)/12.)
C
      X(I+19) =  S302 - R * COS(PI*(I-1)/12.)
      Y(I+19) =  .5  -  R * SIN(PI*(I-1)/12.)
  220 CONTINUE
      X(34) = S302
      Y(34) = - .5
      X(31) = S302
      Y(31) = (Y(26)+Y(34)) / 2.
      X(45) = -.5* TAN(PI/6.)
      Y(45) = .5
      DX = (1.-R)/(2.*COS(PI/6.))
      X(36) = X(45) + DX
      Y(36) = .5
      X(39) = (X(2U)+X(36))/2.
C
      Y(39) = .5
      DO 230 I = 1,4
      X(I+45) = (4 -I)* X(45) /4.
      Y(I+45) = (4 -I)* Y(45) /4.
  230 CONTINUE
C
      DELX = X(46) -X(45)
      DELY = Y(46)-Y(45)
      DO 240 I = 1,8
      X(I+36) = X(I+35) +DELX
      Y(I+36) = Y(I+35) +DELY
  240 CONTINUE
      X(32) = X(44) +(X(34)-X(44))/3.
      Y(32) = -.5
      X(33) = 2.*X(32) - X(44)
      Y(33) = -.5
      X(27) = X(23) + (X(32)-X(23))/3.
      Y(27) = Y(23) + (Y(32)-Y(23))/3.
      X(28) = 2.*X(27)-X(23)
      Y(28) = 2.*Y(27)-Y(23)
```

```
      X(29) = X(28) +(X(31)-X(28))/3.
      Y(29) = Y(28) +(Y(31)-Y(28))/3.
      X(30) = 2.*X(29)-X(28)
      Y(30) = 2.*Y(29)-Y(28)
C
C
      DO 250 I=50,97
      X(I) = -X(98-I)
      Y(I) = -Y(98-I)
  250 CONTINUE
C
C
C   PRINT OUT NODAL DATA
      I = 1
   12 CONTINUE
      LINE = 4
      PRINT 1010
   13 CONTINUE
      PRINT 1011, I, X(I), Y(I), P(I,1), P(I,2), D(I,1), D(I,2),
     1            TH(I), ALF(I)
      I = I+1
      IF(I.GT.KTOTAL)  GO TO 14
      LINE =  LINE + 2
      IF(LINE.GT.56) GO TO 12
      GO TO 13
   14 CONTINUE
 1011 FORMAT(1H  I4,7(3XE11.4),5X,F11.2/)
 1010 FORMAT(5H1NODE,7X,1HX,13X,1HY,13X,2HP1,12X,2HP2,12X,2HD1,12X,2HD2,
     1        8X,9HTHICKNESS,9X,5HALPHA /)
C
C   I       - NUMBER OF NODE
C   X       - X- COORDINATE OF ITH NODE
C   Y       - Y- COORDINATE OF ITH NODE
C   P(I,1)  - KNOWN AND UNKNOWN FORCE COMPONENTS ALONG 1 DIRECTION
C   P(I,2)  - KNOWN AND UNKNOWN FORCE COMPONENTS ALONG 2 DIRECTION
C   D(I,1)  - KNOWN AND UNKNOWN DISPLACEMENTS ALONG 1 DIRECTION
C   D(I,2)  - KNOWN AND UNKNOWN DISPLACEMENTS ALONG 2 DIRECTION
C   ALF(I)  - ANGLE BETWEEN X DIRECTION AND 1 DIRECTION (POSITIVE WHEN
C             COUNTER-CLOCKWISE)
C   TH(I)   - PLATE THICKNESS AT ITH NODE
C
C   GET MSK MATRIX
      JP = 0
      DO 27  J=1,KTOTAL
      DO 27  I=1, 2
      IF (P(J,I).GT.PK)  GO TO 27
      JP = JP+1
      MSK(JP) = 2 * (J-1) + I
   27 CONTINUE
C    MSK  IS MATRIX OF INDICES OF KNOWN FORCES
C    IF FORCE P IS UNKNOWN, IT IS INPUT AS 1000.
C   NOW READ IN TRIANGLE DATA
C
C   JTOTAL - TOTAL NUMBER OF TRIANGLES
      DO 19 I=1,79
      DO 19 J=1,3
   19 L(I,J) = JL(I,J)
      DO 260 I= 80,158
      DO 260 J= 1,3
      L(I,J) = 98 - L(159-I,J)
  260 CONTINUE
C   L(J,1) - INDEX OF THE FIRST  NODE OF THE JTH TRIANGLE
```

```
C  L(J,2) - INDEX OF THE SECOND NODE OF THE JTH TRIANGLE
C  L(J,3) - INDEX OF THE THIRD  NODE OF THE JTH TRIANGLE
C
      DO 20 I=1,JTOTAL
      IF(I.LE.36.OR.I.GE.123) GO TO 300
      EX(I) = EPII
      EY(I) = EPII
      NUXY(I) =XNUPII
      GXY(I) = EPII/(2.*(1.+XNUPII))
      GO TO 310
  300 EX(I) = EPI
      EY(I) = EPI
      NUXY(I) = XNUPI
      GXY(I) = EPI   / (2.*(1.+XNUPI))
  310 CONTINUE
   20 CONTINUE
C  PRINT OUT TRIANGLE DATA
      LINE = 4
      PRINT  1012
      DO 24 I=1,JTOTAL
      IF(LINE.LT.54)  GO TO 22
      LINE = 4
      PRINT 1012
 1012 FORMAT(1H1,8HTRIANGLE,4X6HNODE 1,3X,6HNODE 2,3X,6HNODE 3,18X,2HEX,
     1    18X 2HEY,16X,4HNUYX,17X,3HGXY//)
   22 LINE = LINE+2
   24 PRINT 1023, I, (L(I,J),J=1,3),EX(I),EY(I),NUYX(I),GXY(I)
 1023 FORMAT(1H ,5X,I3,7X,I3,2(6X,I3),4(5X,E15.8)/)
C
C
      MSIZE = 156
      KREM = 6
      N = 25
      M = MSIZE / N
      IF(KREM.NE.N)  M = M+1
C
      LIM1(1) = 1
      LIM1(2) = 1
      LIM1(3) = 1
      LIM1(4) = 1
      LIM1(5) = 1
      LIM1(6) = 1
      LIM1(7) = 1
      LIM2(1) = 158
      LIM2(2) = 158
      LIM2(3) = 158
      LIM2(4) = 158
      LIM2(5) = 158
      LIM2(6) = 158
      LIM2(7) = 158

C
C  PRINT OUT PARTITION INFORMATION
      PRINT 1008
 1008 FORMAT(1H1,12X,20HTRIANGLES CONSIDERED)
      DO 31  I=1,M
      ISIZE = N
      IF(I.EQ.1)  ISIZE = KREM
      PRINT 1009,  I,LIM1(I),LIM2(I),  ISIZE
   31 CONTINUE
 1009 FORMAT(10H PARTITION,6X 5HFIRST,2X,2HTO,2X,4HLAST, 6X,9HDIMENSION/
     1     4X,I3,11X,I3,6X,I3,10X,I3)
```

```
      DO 28 I = 1,KTOTAL
      ALF(I) = ALF(I)/RAD
   28 CONTINUE
C  ALL ANGLES NOW IN RADIANS
      DO 400 I=1, JTOTAL
C  I   IS A TRIANGLE COUNTER
      B11 = 1./EX(I)
      B12 = -NUYX(I) / EY(I)
      B22 = 1. / EY(I)
      B33 = 1. / GXY(I)
      DELTA = B11*B22 -B12**2
      CP(2,1)   = - B12 / DELTA
      CP(1,1)   = B22 / DELTA
      CP(3,1)   =  0.
      CP(1,2) = CP(2,1)
      CP(2,2)   = B11/ DELTA
      CP(3,2)   =  0.
      CP(1,3)   =  0.
      CP(2,3)   =  0.
      CP(3,3) = 1./B33
C
C  C NOW IN C(3,3)   , MATRIX I
C
   30 CONTINUE
      THOMEG   = X(L(I,2))*Y(L(I,3)) +X(L(I,1))*Y(L(I,2))
     1         + Y(L(I,1))*X(L(I,3)) -X(L(I,3))*Y(L(I,2))
     2         - X(L(I,1))*Y(L(I,3)) -Y(L(I,1))*X(L(I,2))
      THOMEG=(TH(L(I,1))+TH(L(I,2))+TH(L(I,3)))/ 6. * THOMEG
      XI2 = X(L(I,2))- X(L(I,1))
      XI3 = X(L(I,3))- X(L(I,1))
      ETA2= Y(L(I,2))- Y(L(I,1))
      ETA3= Y(L(I,3))- Y(L(I,1))
      DELTA = XI2*ETA3 - XI3*ETA2
      DO 38 II = 1,3
      DO 38 JJ = 1,3
      A(II+3,JJ,1) = 0.
      A(II,JJ+3,1) = 0.
      A(II+3,JJ,2) = 0.
      A(II,JJ+3,2) = 0.
   38 CONTINUE
      A(1,1,1) = 1.
      A(2,1,1) = -(ETA3-ETA2) / DELTA
      A(3,1,1) = (XI3 - XI2) / DELTA
      A(1,2,1) = 0.
      A(2,2,1) = ETA3 / DELTA
      A(3,2,1) = -XI3 / DELTA
      A(1,3,1) = 0.
      A(2,3,1) = -ETA2 / DELTA
      A(3,3,1) = XI2 / DELTA
      DO 39 II = 1,3
      DO 39 JJ = 1,3
      A(II+3,JJ+3,1) = A(II,JJ,1)
      A(II,JJ,2) = A(JJ,II,1)
      A(II+3,JJ+3,2) = A(JJ,II,1)
   39 CONTINUE
C
C  TRANSPOSE OF INVERSE OF A NOW IN A(1,1,2) , A INVERSE STILL IN A
C
      CALL MXMULT(DO,A,KMX(1,1),3,6,6)
      CALL MXMULT(C,KMX(1,1)   ,CDA(1,1,I),3,3,6)
C  PRODUCT C*D*(A**-1) NOW IN CDA(1,1,I), ITH TRIANGLE
C
```

```
      CALL MXMULT(A(1,1,2),DTO,A,6,6,3)
      CALL MXCON(A,KPR(1,1,I),THOMEG,6,3)
      CALL MXMULT(KPR(1,1,I),CDA(1,1,I),KMX,6,3,6)
C
C  MATRIX K(I) NOW IN KMX, TRIANGLE I
      DO 40 II =1,6
      DO 40 JJ =1,6
      T(II,JJ)    = 0.
   40 CONTINUE
      T(1,1)    =  COS(ALF(L(I,1)))
      T(4,1)    =  SIN(ALF(L(I,1)))
      T(2,2)    =  COS(ALF(L(I,2)))
      T(5,2)    =  SIN(ALF(L(I,2)))
      T(3,3)    =  COS(ALF(L(I,3)))
      T(6,3)    =  SIN(ALF(L(I,3)))
      T(1,4)    = -T(4,1)
      T(4,4)    =  T(1,1)
      T(2,5)    = -T(5,2)
      T(5,5)    =  T(2,2)
      T(3,6)    = -T(6,3)
      T(6,6)    =  T(3,3)
      CALL MXMULT (KMX,T,A,6,6,6)
      T(4,1) = - T(4,1)
      T(5,2) = - T(5,2)
      T(6,3) = - T(6,3)
      T(1,4) = - T(1,4)
      T(2,5) = - T(2,5)
      T(3,6) = - T(3,6)
C
C  INVERSE OF T NOW IN T
C
      CALL MXMULT (T,A,KPR(1,1,I),6,6,6)
C
C  K-PRIME NOW IN KPR , A  HAS BEEN CLOBBERED.
  400 CONTINUE
      RETURN
  800 PRINT 1051,II,JJ,II
 1051 FORMAT(1H1, 5H  EF(,I3,1H,I3,7H) = ER(,I3,6H) = 0.)
      STOP
  900 CONTINUE
      PRINT 1050    , J
 1050 FORMAT(1H1  35HCOULD NOT INVERT MATRIX  T1,TRIANGLE,I3)
  950 CONTINUE
      REWIND 20
      STOP
      END
```

CASE 2

```
      SUBROUTINE STRESS
C
C      STRESS SUBROUTINE CASE 2
C
C
C  THIS SUBROUTINE DERIVES AND PRINTS STRESSES
C
      COMMON /STRSS/ ALF(155), MSIZE, R, GAMMA, EI, GII,XNU(2)
      COMMON /1/  S(32,32,5),L(300,3),G(32),SPACE(20)
     1          , NUF(3,314)        ,X(155),Y(155),CDA(3,6,250)
     2          , KPR(6,6,250),MSK(310),JTOTAL,N,KREM,M , IMR, IPL
     3          ,  P(155,2)   ,D(155,2)
      COMMON/LIM/  LIM1(10),LIM2(10)
      DIMENSION  DVX(6), SIGOUT(632)
      DIMENSION    SP(32,10)
      EQUIVALENCE(S,SIGOUT) ,(NUF,SP)
      TYPE REAL KPR
      EQUIVALENCE  (S,SIG) , (S(931),PSTR) , (S(1861),X0),(S(2161),Y0)
      EQUIVALENCE  (S(2461),DEL) , (S(2761),DX)
      DIMENSION    ERR(310)
      DIMENSION    DX(6) , SIG(4,168)  ,   PSTR(3,310)
      DIMENSION   X0(300) , Y0(300), DEL(310)
      DIMENSION   KS22(32,96)
      EQUIVALENCE    (KPR,KS22)
      TYPE REAL KS22
      DATA (PK=999.)
C  REMOVE GAPS FROM SP(32,10) = DEL(310)
      DO 5 J=1,KREM
      DEL(J) = SP(J,1)
    5 CONTINUE
      KLOC = KREM -N
      DO 10 I = 2,M
      KLOC = KLOC + N
      DO 10 J = 1,N
      DEL(KLOC+J) = SP(J,I)
   10 CONTINUE
      PRINT 1010
 1010 FORMAT(1H1,50X,13HDISPLACEMENTS,//)
      NDEL = MSIZE/7
      JCNT = 0
      DO 15 J=1,NDEL
      JCNT = JCNT + 1
      IF(JCNT.LE.18)  GO TO 14
      PRINT 1010
      JCNT = 0
   14 JFIR = 7*(J-1) + 1
      JLAST =  JFIR + 6
      PRINT 1011 , (K,K=JFIR,JLAST)
 1011 FORMAT(1H ,7(8X,4HDEL(,I3,1H) ))
      PRINT 1012, (DEL(K),K=JFIR,JLAST)
 1012 FORMAT(1H ,7(2X,E14.7)/)
   15 CONTINUE
      LOC1 = 7*NDEL+1
      LOC2 = MSIZE
      IF(LOC1.GT.LOC2)  GO TO 20
      PRINT 1011,(K,K=LOC1,LOC2)
      PRINT 1012, (DEL(K),K=LOC1,LOC2)
   20 CONTINUE
      DO 855  I=1,JTOTAL
      J = I
      KZ = 0
      DO 831 KK =1,3
```

```
      DO 832 KJ =1,2
      KZ = K   + 1
      IF( PL(J,KK),KJ).GT.PK) GO TO 828
      IF(2*(L(J,KK)-1)+KJ-MSK(IPL)) 801, 802, 803
  802 II2 = IPL
      GO TO 827
  801 IMR = IPL - 1
  806 IF(2*(L(J,KK)-1) +KJ -MSK(IMR))  804, 805, 805
  805 II2 = IMR
      IPL = IMR
      GO TO 827
  804 IMR = IMR - 1
      GO TO 806
  803 IMR = IPL + 1
  807 IF(2*(L(J,KK)-1) +KJ -MSK(IMR)) 805, 805, 810
  810 IMR = IMR + 1
      GO TO 807
  827 DVX(KZ) = DEL(II2)
      GO TO 832
  828 DVX(KZ)= D(L(J,KK),KJ)
  832 CONTINUE
      DX(KZ-1) = DVX(KZ-1)*COS(ALF(L(J,KK)))-DVX(KZ)*SIN(ALF(L(J,KK)))
      DX(KZ) = DVX(KZ-1)*SIN(ALF(L(J,KK)))+DVX(KZ)*COS(ALF(L(J,KK)))
  831 CONTINUE
      DX2 = DX(2)
      DX(2) = DX(3)
      DX(3) = DX(5)
      DX(5) = DX(4)
      DX(4) = DX2
      CALL MXMULT(CDA(1,1,I),DX, SIG(1,I),3,6,1)
      KK3 = 1
      IF(I.GE.37.AND.I.LE.122)  KK3 = 2
      SIG(4,I) = SIG(3,I)
      SIG(3,I) = XNU(KK3)*(SIG(1,I)+SIG(2,I))
C   SIGMA NOW IN SIG(1,I)   , TRIANGLE I
      XO(I) = 0.
      YO(I) = 0.
      DO 840  K = 1,3
      XO(I) = XO(I) + X(L(I,K))
      YO(I) = YO(I) + Y(L(I,K))
  840 CONTINUE
      XO(I) = XO(I) / 3.
      YO(I) = YO(I) / 3.
  850 CONTINUE
  855 CONTINUE
      ENBARY = R/4.*(SIG(2,1)+SIG(2,4)+SIG(2,13)+SIG(2,25))
     1      +(X(20)-X(35))*(SIG(2,37)+SIG(2,50))
     2    +(X(36)-X(45))*(SIG(2,64)+SIG(2,80))
     3     +(X(54)-X(64))/3.*(SIG(2,96)+SIG(2,111)+SIG(2,113))
      ENBARY = SQRT(3.)/ENBARY
      DO 860 I=1,632
  860 SIGOUT(I) = SIGOUT(I)*ENBARY
      I = 1
  870 CONTINUE
      PRINT 1000
      LINE = 3
 1000 FORMAT(1H1,2(8HTRIANGLE,6X,8HCENTROID,9X,12HVECTOR SIGMA,10X)/)
  871 CONTINUE
      II = I+1
      DO  880  J=1,4
      GO TO (873,874,875,875), J
  873 PRINT 1001, I, XO(I), SIG(J,I),II,XO(II), SIG(J,II)
```

```
      GO TO 878
  874 PRINT 1002,    YO(I),SIG(J,I),      YO(II), SIG(J,II)
      GO TO 878
  875 PRINT 1003, SIG(J,I), SIG(J,II)
  878 CONTINUE
  880 CONTINUE
      LINE = LINE + 6
      I = I+ 2
      PRINT 1004
 1001 FORMAT(1H 2(5X,I3,F14.6, 7X,E15.8,10X))
 1002 FORMAT(1H 2(8X,F14.6, 7X,E15.8,10X) )
 1003 FORMAT(1H 2(29X,E15.8,10X))
 1004 FORMAT(/)
 1069 FORMAT(1H ,30X,E15.8)
      IF(I.GT.JTOTAL)  GO TO 890
      IF (LINE.GT.54) GO TO 870
      GO TO 871
  890 CONTINUE
      WRITE (20)  (SIGOUT(I),I=1,316)                                  CASE 2
      PRINT 8787, ENBARY
 8787 FORMAT(//23H NORMALIZATION FACTOR =,2X,E15.7)
      RETURN
      END
```

```
      PROGRAM MAIN
C
C  THIS IS THE EXECUTIVE PROGRAM USED WITH SUBROUTINE BIGMX AND SUBROUTINE
C  CHLSKY  TO GENERATE AND SOLVE LARGE SYSTEMS OF LINEAR EQUATIONS
C
      REWIND 20                                                        CASE 6
      CALL TAPESKIP(20,6,0)
      CALL TAPESKIP(20,3,0)
    1 CONTINUE
      CALL INPUT
C
C  NOW HAVE ALL K-PRIME AND CDA MATRICES
C
      CALL CHLSKY
C
C  NOW HAVE SOLUTION U IN SP
C
      CALL STRESS
C
C  ALL STRESSES NOW PRINTED OUT
C
      GO TO 1
                                                                     22 CARDS
```

```
      SUBROUTINE INPUT
C
C     TRANSVERSE SHEAR    CASE 6
C
C
C    THIS SUBROUTINE READS AND PRINTS THE INPUTS FOR THE PLANAR FINITE
C   ELEMENT PROGRAM.  ALL INPUTS NOT READ ARE GENERATED HERE.
C
      COMMON /STRSS/ ALF(155), MSIZE, R, GAMMA, EI, GII,XNU(2)
      COMMON /1/  S(32,32,5),L(300,3),G(32),SPACE(20)
     1           , NUF(3,314)          ,X(155),Y(155),CDA(3,6,250)
     2          , KPR(6,6,250),MSK(310),JTOTAL,N,KREM,M , IMR, IPL
     3           ,  P(155,2)   ,D(155,2)
      COMMON/LIM/  LIM1(10),LIM2(10)
      DIMENSION    SP(32,10)
      DIMENSION NUYX(250),NUXY(250),   GXY(250), EX(250), EY(250)
      DIMENSION BUMP(10) ; BUMP1(5)
      EQUIVALENCE  (NUF,SP)
      EQUIVALENCE  (C,CP)
      DIMENSION JL(79,3)
      DIMENSION COMENT(10)
C
C  NUYX = NUXY, THIS MODIFICATION
      EQUIVALENCE  (NUXY,NUYX)
      DIMENSION  TH(155) , CP(3,3) , T1(3,3,2),T2(3,3),DTD(18),
     1  T(6,6),KMX(6,6), C(3,3),A(6,6,2),DO(3,6),DD(18),DTO(6,3)
      EQUIVALENCE  (S,F) , (S(901),VF), (S(1801),PHI),(S(2701),ER)
     1           , (S(3001),EF),(S(3901),NUR)
     2           ,  (DO,DD) , (DTO,DTD)   , (S(4356),TH)
     3        ,  (S(4511),CP) , (S(4521),T1), (S(4541),T2)
     4        ,  (S(4551),CABC) , (S(4581),T) , (S(4621),KMX)
     5     ,    (S(4671),A)
      TYPE REAL NUXY
      TYPE INTEGER  BUMP, BUMP1
      TYPE REAL     NUYX, NUR, NUF, KPR, KMX
      DATA  (KTOTAL = 97)
      DATA(BUMP=  2,4,7,7,15,1 ,9 ,9 ,12,-1)
      DATA(BUMP1= 5,7,7,7,5 )   ,(PI= 3.1415927)
      DATA  (RAD = 57.29578)
      DATA  (XLIM=1.E-8)
      DATA (PK=999.)
      DATA ((DD(JJ),JJ=1,18) = 0., 0., 0., 1., 0., 0., 0., 0., 1.,
     1                         0., 0., 0., 0., 0., 1., 0., 1., 0. )
      DATA ((DTD(JJ),JJ=1,18) = 0., 1., 0., 0., 0., 0., 0., 0., 0.,
     1                          0., 0., 1., 0., 0., 1., 0., 1., 0.)
      DATA(((JL(I,J),I=1,79),J=1,3) =
     1   1,1,1,2,2,3,3,3,4,4,4,5,6,7,7,8,8,8,9,10,10,11,11,12,13,13,
     2   14,15,15,16,16,17,17,18,18,19,20,23,24,24,25,25,25,26,28,29,
     3   29,30,30,35,21,21,21,21,22,22,23,23,27,27,27,28,28,36,46,37,
     4   47,38,48,39,39,40,41,41,41,42,42,43,43,
     5   2,3,4,6,7,7,8,9,9,10,11,11,13,6,14,14,15,16,16,16,17,10,
     6   18,18,20,21,21,21,22,15,23,23,24,17,25,25,35,27,27,28,28,29,
     7   30,30,32,32,33,33,34,36,35,37,38,39,39,40,40,41,41,42,43,43,44,
     8   45,37,46,38,47,39,48,49,49,40,50,51,51,52,52,53,
     9   3,4,5,7,3,8,9,4,10,11,5,12,14,14,8,15,16,9,10,17,18,18,12,
     1   19,21,14,15,22,23,23,17,24,25,25,19,26,21,24,28,25,29,30,26,
     2   31,29,33,30,34,31,37,37,38,39,22,40,23,41,27,42,43,28,44,32,
     3   46,36,47,37,48,38,49,40,50,50,51,42,52,43,53,44 )
C
      READ 1000,(COMENT(I),I=1,10)
 1000 FORMAT(10A8)
      PRINT 1001,(COMENT(I),I=1,10)
```

```
 1001 FORMAT(1H1,10A8)
C
C  KTOTAL - TOTAL NUMBER OF NODES CONSIDERED
C
C
C  READ AND PRINT
C
      JTOTAL = 158
      READ 1003, VF, GAMMA,EI, XNUI, BETA
 1003 FORMAT(5E10.4)
      IF (VF.EQ.0.)  GO TO 950
      PRINT 1018
      PRINT 1016
      PRINT 1017,  VF, GAMMA, EI, XNUI, BETA
 1016 FORMAT(1H ,13X,2HVF, 10X,5HGAMMA, 13X,2HEI, 11X,4HXNUI, 11X4HBETA)
 1017 FORMAT(1H ,5(E15.8))
 1018 FORMAT(////)

      S3 = SQRT(3.)
      S302 = S3/2.
      EII = EI / GAMMA
      XNUII = XNUI/BETA
      GII = EII/(2.*(1.+XNUII))
      XNU(1) = XNUI
      XNU(2) = XNUII
C
      EPI = EI
      EPII = EII
      XNUPI = XNUI
      XNUPII = XNUII
      EPI = EI/(1.-XNUI**2)
      EPII = EII/(1.-XNUII**2)
C
      XNUPI = XNUI/(1.-XNUI)
      XNUPII = XNUII/(1.-XNUII)
      DO 200 I = 1,97
      TH(I) =1.
      ALF(I)=0.
      D(I,1) = 1000.
      D(I,2) = 1000.
      P(I,1) = 0.
      P(I,2) = 0.
  200 CONTINUE
      I = 0
      DO 201  J= 1,10
      I = I +BUMP(J)
      D(I,1) = 0.
      D(98-I,1) = 0.
      P(I,1) = 1000.
      P(98-I,1) = 1000.
  201 CONTINUE
      I = 0
      DO  202 J = 1,5
      I = I+ BUMP1(J)
      D(I,2) = 1.
      D(98-I,2) = -1.
      P(I,2) = 1000.
      P(98-I,2) = 1000.
  202 CONTINUE
      D(1,1) = 0.
      D(1,2) = 1.
      P(1,1) = 1000.
```

```
      P(1,2) = 1000.
      D(34,1) = 0.
      D(34,2) = 1.
      P(34,1) = 1000.
      P(34,2) = 1000.
      D(64,1)  = 0.
      D(64,2) = -1.
      P(64,1) = 1000.
      P(64,2) = 1000.
      D(97,1) = 0.
      D(97,2) = -1.
      P(97,1) = 1000.
      P(97,2) = 1000.
      P(64,2) = 1000.
      R= SQRT(2.*S3*VF/PI)
      X(1) = S302
      Y(1) = .5
      DO 210 I=1,4
      X(I+1) = S302 - R/4.*COS(PI*(I-1)/6.)
      Y(I+1) = .5 -  R/4. *SIN(PI*(I-1)/6.)
  210 CONTINUE
C
      DO 220  I=1,7
      X(I+5) = S302 - R/2.* COS(PI*(I-1)/12.)
      Y(I+5) = .5    - R/2.* SIN(PI*(I-1)/12.)
C
      X(I+12) =  S302 - 3.*R/4.*COS(PI*(I-1)/12.)
      Y(I+12) =  .5  -  3.*R/4.*SIN(PI*(I-1)/12.)
C
      X(I+19) =  S302 - R * COS(PI*(I-1)/12.)
      Y(I+19) =  .5  -  R * SIN(PI*(I-1)/12.)
  220 CONTINUE
      X(34) = S302
      Y(34) = - .5
      X(31) = S302
      Y(31) = (Y(26)+Y(34)) / 2.
      X(45) = -.5* TAN(PI/6.)
      Y(45) = .5
      DX = (1.-R)/(2.*COS(PI/6.))
      X(36) = X(45) + DX
      Y(36) = .5
      X(35) = (X(20)+X(36))/2.
      Y(35) = .5
C
      DO 230 I = 1,4
      X(I+45) = (4 -I)* X(45) /4.
      Y(I+45) = (4 -I)* Y(45) /4.
  230 CONTINUE
C
      DELX = X(46) -X(45)
      DELY = Y(46)-Y(45)
      DO 240 I = 1,8
      X(I+36) = X(I+35) +DELX
      Y(I+36) = Y(I+35) +DELY
  240 CONTINUE
      X(32) = X(44) +(X(34)-X(44))/3.
      Y(32) = -.5
      X(33) = 2.*X(32) - X(44)
      Y(33) = -.5
      X(27) = X(23) + (X(32)-X(23))/3.
      Y(27) = Y(23) + (Y(32)-Y(23))/3.
      X(28) = 2.*X(27)-X(23)
```

```
      Y(28) = 2.*Y(27)-Y(23)
      X(29) = X(28) +(X(31)-X(28))/3.
      Y(29) = Y(28) +(Y(31)-Y(28))/3.
      X(30) = 2.*X(29)-X(28)
      Y(30) = 2.*Y(29)-Y(28)
C
C
      DO 250 I=50,97
      X(I) = -X(98-I)
      Y(I) = -Y(98-I)
  250 CONTINUE
C
C
C  PRINT OUT NODAL DATA
      I = 1
   12 CONTINUE
      LINE = 4
      PRINT 1010
   13 CONTINUE
      PRINT 1011, I, X(I), Y(I), P(I,1), P(I,2), D(I,1), D(I,2),
     1             TH(I), ALF(I)
      I = I+1
      IF(I.GT.KTOTAL)  GO TO 14
      LINE =  LINE + 2
      IF(LINE.GT.56) GO TO 12
      GO TO 13
   14 CONTINUE
 1011 FORMAT(1H  I4,7(3XE11.4),5X,F11.2/)
 1010 FORMAT(5H1NODE,7X,1HX,13X,1HY,13X,2HP1,12X,2HP2,12X,2HD1,12X,2HD2,
     1        8X,9HTHICKNESS,9X,5HALPHA /)
C
C  I       - NUMBER OF NODE
C  X       - X- COORDINATE OF ITH NODE
C  Y       - Y- COORDINATE OF ITH NODE
C  P(I,1) - KNOWN AND UNKNOWN FORCE COMPONENTS ALONG 1 DIRECTION
C  P(I,2) - KNOWN AND UNKNOWN FORCE COMPONENTS ALONG 2 DIRECTION
C  D(I,1) - KNOWN AND UNKNOWN DISPLACEMENTS ALONG 1 DIRECTION
C  D(I,2) - KNOWN AND UNKNOWN DISPLACEMENTS ALONG 2 DIRECTION
C  ALF(I) - ANGLE BETWEEN X DIRECTION AND 1 DIRECTION (POSITIVE WHEN
C           COUNTER-CLOCKWISE)
C  TH(I)  - PLATE THICKNESS AT ITH NODE
C
C  GET MSK MATRIX
      JP = 0
      DO 27  J=1,KTOTAL
      DO 27  I=1, 2
      IF (P(J,I).GT.PK)  GO TO 27
      JP = JP+1
      MSK(JP) = 2 * (J-1) + I
   27 CONTINUE
C   MSK  IS MATRIX OF INDICES OF KNOWN FORCES
C   IF FORCE P IS UNKNOWN, IT IS INPUT AS 1000.
C  NOW READ IN TRIANGLE DATA
C
C  JTOTAL - TOTAL NUMBER OF TRIANGLES
      DO 19 I=1,79
      DO 19 J=1,3
   19 L(I,J) = JL(I,J)
      DO 260 I= 80,158
      DO 260 J= 1,3
      L(I,J) = 98 - L(159-I,J)
  260 CONTINUE
```

```
C  L(J,1) - INDEX OF THE FIRST  NODE OF THE JTH TRIANGLE
C  L(J,2) - INDEX OF THE SECOND NODE OF THE JTH TRIANGLE
C  L(J,3) - INDEX OF THE THIRD  NODE OF THE JTH TRIANGLE
C
      DO 20 I=1,JTOTAL
      IF(I.LE.36.OR.I.GE.123) GO TO 300
      EX(I) = EPII
      EY(I) = EPII
      NUXY(I) =XNUPII
      GXY(I) = EPII/(2.*(1.+XNUPII))
      GO TO 310
  300 EX(I) = EPI
      EY(I) = EPI
      NUXY(I) = XNUPI
      GXY(I) = EPI  / (2.*(1.+XNUPI))
  310 CONTINUE
   20 CONTINUE
C  PRINT OUT TRIANGLE DATA
      LINE = 4
      PRINT  1012
      DO 24 I=1,JTOTAL
      IF(LINE.LT.54)  GO TO 22
      LINE = 4
      PRINT 1012
 1012 FORMAT(1H1,8HTRIANGLE,4X6HNODE 1,3X,6HNODE 2,3X,6HNODE 3,18X,2HEX,
     1   18X 2HEY,16X,4HNUYX,17X,3HGXY//)
   22 LINE = LINE+2
   24 PRINT 1023, I, (L(I,J),J=1,3),EX(I),EY(I),NUYX(I),GXY(I)
 1023 FORMAT(1H ,5X,I3,7X,I3,2(6X,I3),4(5X,E15.8)/)
C
C
      MSIZE = 156
      KREM = 6
      N = 25
      M = MSIZE / N
      IF(KREM.NE.N)  M = M+1
C
      LIM1(1) = 1
      LIM1(2) = 1
      LIM1(3) = 1
      LIM1(4) = 1
      LIM1(5) = 1
      LIM1(6) = 1
      LIM1(7) = 1
      LIM2(1) = 158
      LIM2(2) = 158
      LIM2(3) = 158
      LIM2(4) = 158
      LIM2(5) = 158
      LIM2(6) = 158
      LIM2(7) = 158

C
C  PRINT OUT PARTITION INFORMATION
      PRINT 1008
 1008 FORMAT(1H1,12X,20HTRIANGLES CONSIDERED)
      DO 31  I=1,M
      ISIZE = N
      IF(I.EQ.1)  ISIZE = KREM
      PRINT 1009,  I,LIM1(I),LIM2(I),  ISIZE
   31 CONTINUE
 1009 FORMAT(10H PARTITION,6X 5HFIRST,2X,2HTO,2X,4HLAST, 6X,9HDIMENSION/
```

```
     1    4X,I3,11X,I3,6X,I3,10X,I3)
      DO 28 I = 1,KTOTAL
      ALF(I) = ALF(I)/RAD
   28 CONTINUE
C   ALL ANGLES NOW IN RADIANS
      DO 400 I=1, JTOTAL
C   I   IS A TRIANGLE COUNTER
      B11 = 1./EX(I)
      B12 = -NUYX(I) / EY(I)
      B22 = 1. / EY(I)
      B33 = 1. / GXY(I)
      DELTA = B11*B22 -B12**2
      CP(2,1)    = - B12 / DELTA
      CP(1,1)    = B22 / DELTA
      CP(3,1)    =  0.
      CP(1,2) = CP(2,1)
      CP(2,2)    = B11/ DELTA
      CP(3,2)    =  0.
      CP(1,3)    =  0.
      CP(2,3)    =  0.
      CP(3,3) = 1./B33
C
C   C NOW IN C(3,3)   , MATRIX I
C
   30 CONTINUE
      THOMEG    = X(L(I,2))*Y(L(I,3)) +X(L(I,1))*Y(L(I,2))
     1          + Y(L(I,1))*X(L(I,3)) -X(L(I,3))*Y(L(I,2))
     2          - X(L(I,1))*Y(L(I,3)) -Y(L(I,1))*X(L(I,2))
      THOMEG=(TH(L(I,1))+TH(L(I,2))+TH(L(I,3)))/ 6. * THOMEG
      XI2 = X(L(I,2))- X(L(I,1))
      XI3 = X(L(I,3))- X(L(I,1))
      ETA2= Y(L(I,2))- Y(L(I,1))
      ETA3= Y(L(I,3))- Y(L(I,1))
      DELTA = XI2*ETA3 - XI3*ETA2
      DO 38 II = 1,3
      DO 38 JJ = 1,3
      A(II+3,JJ,1) = 0.
      A(II,JJ+3,1) = 0.
      A(II+3,JJ,2) = 0.
      A(II,JJ+3,2) = 0.
   38 CONTINUE
      A(1,1,1) = 1.
      A(2,1,1) = -(ETA3-ETA2) / DELTA
      A(3,1,1) = (XI3 - XI2) / DELTA
      A(1,2,1) = 0.
      A(2,2,1) = ETA3 / DELTA
      A(3,2,1) = -XI3 / DELTA
      A(1,3,1) = 0.
      A(2,3,1) = -ETA2 / DELTA
      A(3,3,1) = XI2 / DELTA
      DO 39 II = 1,3
      DO 39 JJ = 1,3
      A(II+3,JJ+3,1) = A(II,JJ,1)
      A(II,JJ,2) = A(JJ,II,1)
      A(II+3,JJ+3,2) = A(JJ,II,1)
   39 CONTINUE
C
C   TRANSPOSE OF INVERSE OF A NOW IN A(1,1,2) , A INVERSE STILL IN A
C
      CALL MXMULT(DO,A,KMX(1,1),3,6,6)
      CALL MXMULT(C,KMX(1,1)  ,CDA(1,1,I),3,3,6)
C   PRODUCT C*D*(A**-1) NOW IN CDA(1,1,I), ITH TRIANGLE
```

```
C
      CALL MXMULT(A(1,1,2),DTO,A,6,6,3)
      CALL MXCON(A,KPR(1,1,I),THOMFG,6,3)
      CALL MXMULT(KPR(1,1,I),CUA(1,1,I),KMX,6,3,6)
C
C  MATRIX K(I) NOW IN KMX, TRIANGLE I
      DO 40 II =1,6
      DO 40 JJ =1,6
      T(II,JJ)    = 0.
   40 CONTINUE
      T(1,1)    =  COS(ALF(L(I,1)))
      T(4,1)    =  SIN(ALF(L(I,1)))
      T(2,2)    =  COS(ALF(L(I,2)))
      T(5,2)    =  SIN(ALF(L(I,2)))
      T(3,3)    =  COS(ALF(L(I,3)))
      T(6,3)    =  SIN(ALF(L(I,3)))
      T(1,4)    = -T(4,1)
      T(4,4)    =  T(1,1)
      T(2,5)    = -T(5,2)
      T(5,5)    =  T(2,2)
      T(3,6)    = -T(6,3)
      T(6,6)    =  T(3,3)
      CALL MXMULT (KMX,T,A,6,6,6)
      T(4,1) = - T(4,1)
      T(5,2) = - T(5,2)
      T(6,3) = - T(6,3)
      T(1,4) = - T(1,4)
      T(2,5) = - T(2,5)
      T(3,6) = - T(3,6)
C
C  INVERSE OF T NOW IN T
C
      CALL MXMULT (T,A,KPR(1,1,I),6,6,6)
C
C  K-PRIME NOW IN KPR , A  HAS BEEN CLOBBERED.
  400 CONTINUE
      RETURN
  800 PRINT 1051,II,JJ,II
 1051 FORMAT(1H1, 5H  EF(,I3,1H,I3,7H) = ER(,I3,6H) = 0.)
      STOP
  900 CONTINUE
      PRINT 1050    , J
 1050 FORMAT(1H1  35HCOULD NOT INVERT MATRIX  T1,TRIANGLE,I3)
  950 CONTINUE
      REWIND 20                                                    CASE 6
      STOP
      END
```

FUNDAMENTAL CASE 6

```
      SUBROUTINE STRESS
C
C     STRESS SUBROUTINE   CASE 6
C
C
C   THIS SUBROUTINE DERIVES AND PRINTS STRESSES
C
      COMMON /STRSS/ ALF(155), MSIZE, R, GAMMA, EI, GII,XNU(2)
      COMMON /1/  S(32,32,5),L(300,3),G(32),SPACE(20)
     1          , NUF(3,314)        ,X(155),Y(155),CDA(3,6,250)
     2          , KPR(6,6,250),MSK(310),JTOTAL,N,KREM,M , IMR, IPL
     3          ,  P(155,2)   ,D(155,2)
      COMMON/LIM/  LIM1(10),LIM2(10)
      DIMENSION  DVX(6), SIGOUT(632)
      DIMENSION   SP(32,10)
      EQUIVALENCE(S,SIGOUT) ,(NUF,SP)
      TYPE REAL KPR
      EQUIVALENCE  (S,SIG) , (S(931),PSTR) , (S(1861),X0),(S(2161),Y0)
      EQUIVALENCE  (S(2461),DEL) , (S(2761),DX)
      DIMENSION   ERR(310)
      DIMENSION   DX(6) , SIG(4,168)  ,  PSTR(3,310)
      DIMENSION  X0(300) , Y0(300), DEL(310)
      DIMENSION  KS22(32,96)
      EQUIVALENCE   (KPR,KS22)
      TYPE REAL KS22
      DATA (PK=999.)
C  REMOVE GAPS FROM SP(32,10) = DEL(310)
      DO 5 J=1,KREM
      DEL(J) = SP(J,1)
    5 CONTINUE
      KLOC = KREM -N
      DO 10 I = 2,M
      KLOC = KLOC + N
      DO 10 J = 1,N
      DEL(KLOC+J) = SP(J,I)
   10 CONTINUE
      PRINT 1010
 1010 FORMAT(1H1,50X,13HDISPLACEMENTS,//)
      NDEL = MSIZE/7
      JCNT = 0
      DO 15 J=1,NDEL
      JCNT = JCNT + 1
      IF(JCNT.LE.18)  GO TO 14
      PRINT 1010
      JCNT = 0
   14 JFIR = 7*(J-1) + 1
      JLAST =  JFIR + 6
      PRINT 1011 , (K,K=JFIR,JLAST)
 1011 FORMAT(1H ,7(8X,4HDEL(,I3,1H) ))
      PRINT 1012, (DEL(K),K=JFIR,JLAST)
 1012 FORMAT(1H ,7(2X,E14.7)/)
   15 CONTINUE
      LOC1 = 7*NDEL+1
      LOC2 = MSIZE
      IF(LOC1.GT.LOC2)  GO TO 20
      PRINT 1011,(K,K=LOC1,LOC2)
      PRINT 1012, (DEL(K),K=LOC1,LOC2)
   20 CONTINUE
      DO 855  I=1,JTOTAL
      J = I
      KZ = 0
      DO 831 KK =1,3
```

```
      DO 832 KJ =1,2
      KZ = KZ + 1
      IF( P(L(J,KK),KJ).GT.PK) GO TO 828
      IF(2*(L(J,KK)-1)+KJ-MSK(IPL)) 801, 802, 803
  802 II2 = IPL
      GO TO 827
  801 IMR = IPL - 1
  806 IF(2*(L(J,KK)-1) +KJ -MSK(IMR))  804, 805, 805
  805 II2 = IMR
      IPL = IMR
      GO TO 827
  804 IMR = IMR - 1
      GO TO 806
  803 IMR = IPL + 1
  807 IF(2*(L(J,KK)-1) +KJ -MSK(IMR)) 805, 805, 810
  810 IMR = IMR + 1
      GO TO 807
  827 DVX(KZ) = DEL(II2)
      GO TO 832
  828 DVX(KZ)= D(L(J,KK),KJ)
  832 CONTINUE
      DX(KZ-1) = DVX(KZ-1)*COS(ALF(L(J,KK)))-DVX(KZ)*SIN(ALF(L(J,KK)))
      DX(KZ) = DVX(KZ-1)*SIN(ALF(L(J,KK)))+DVX(KZ)*COS(ALF(L(J,KK)))
  831 CONTINUE
      DX2 = DX(2)
      DX(2) = DX(3)
      DX(3) = DX(5)
      DX(5) = DX(4)
      DX(4) = DX2
      CALL MXMULT(CDA(1,1,I),DX, SIG(1,I),3,6,1)
      KK3 = 1
      IF(I.GE.37.AND.I.LE.122)  KK3 = 2
      SIG(4,I) = SIG(3,I)
      SIG(3,I) = XNU(KK3)*(SIG(1,I)+SIG(2,I))
C   SIGMA NOW IN SIG(1,I)   , TRIANGLE I
      XO(I) = 0.
      YO(I) = 0.
      DO 840  K = 1,3
      XO(I) = XO(I) + X(L(I,K))
      YO(I) = YO(I) + Y(L(I,K))
  840 CONTINUE
      XO(I) = XO(I) / 3.
      YO(I) = YO(I) / 3.
  850 CONTINUE
  855 CONTINUE
      TAUBAR=R/4.*(SIG(4,3)+SIG(4,12)+SIG(4,24)+SIG(4,36))
     1      +(1.-R)/2.*(SIG(4,44)+SIG(4,49))
      TAUBAR = 1./TAUBAR
      DO 860 I=1,632
  860 SIGOUT(I)=SIGOUT(I)*TAUBAR
      I = 1
  870 CONTINUE
      PRINT 1000
      LINE = 3
 1000 FORMAT(1H1,2(8HTRIANGLE,6X,8HCENTROID,9X,12HVECTOR SIGMA,10X)/)
  871 CONTINUE
      II = I+1
      DO  880  J=1,4
      GO TO (873,874,875,875), J
  873 PRINT 1001, I, XO(I), SIG(J,I),II,XO(II), SIG(J,II)
      GO TO 878
  874 PRINT 1002,    YO(I),SIG(J,I),      YO(II), SIG(J,II)
```

```
      GO TO 878
  875 PRINT 1003, SIG(J,I), SIG(J,II)
  878 CONTINUE
  880 CONTINUE
      LINE = LINE + 6
      I = I+ 2
      PRINT 1004
 1001 FORMAT(1H 2(5X,I3,F14.6, 7X,E15.8,10X))
 1002 FORMAT(1H 2(8X,F14.6, 7X,E15.8,10X) )
 1003 FORMAT(1H 2(29X,E15.8,10X))
 1004 FORMAT(/)
 1069 FORMAT(1H ,30X,E15.8)
      IF(I.GT.JTOTAL)  GO TO 890
      IF (LINE.GT.54) GO TO 870
      GO TO 871
  890 CONTINUE
      WRITE (20) (SIGOUT(I),I=1,316)                                    CASE 6
      PRINT 8787, TAUBAR
 8787 FORMAT(//23H NORMALIZATION FACTOR =,2X,E15.7)
      RETURN
      END
```

AUXILIARIES

```
      SUBROUTINE BIGMX(ICYCLE)
C
C  THIS SUBROUTINE SEARCHES THE K-PRIME MATRICES BETWEEN LIM1(ICYCLE)
C  AND LIM2(ICYCLE) TO SET UP THE KREM OR N  ROWS OF S=K*22 REQUIRED
C  BY THE EQUATION SOLVING ROUTINE CHLSKY
C
      COMMON /1/  S(32,32,5),L(300,3),G(32),SPACE(20)
     1          , NUF(3,314)        ,X(155),Y(155),CDA(3,6,250)
     2          , KPR(6,6,250),MSK(310),JTOTAL,N,KREM,M , IMR, IPL
     3          ,  P(155,2)   ,D(155,2)
      COMMON/LIM/  LIM1(10),LIM2(10)
      DIMENSION   SP(32,10)
      EQUIVALENCE  (NUF,SP)
      DIMENSION  KS22(32,96)     ,  KS21(32)
      DIMENSION  Q(5120)
      EQUIVALENCE  (S,KS22,Q)   , (G,KS21)
      TYPE REAL KPR
      TYPE REAL  KS22 , KS21
      DATA  (PK = 999.)
      IF(ICYCLE.GT.1) GO TO 1
      ISHIFT =0
      JSHIFT =0
      K1 = KREM
      K2 = KREM + N
      GO TO 109
    1 CONTINUE
      IF (ICYCLE.EQ.M)  GO TO 3
      K1 = N
      K2 = 3*N
      GO TO  2
    3 CONTINUE
      K1 = N
      K2 = 2*N
    2 CONTINUE
      IF(ICYCLE.GT.2)  GO TO 110
      JSHIFT = 0
      GO TO 111
  110 CONTINUE
      JSHIFT = KREM + (ICYCLE-3)*N
  111 CONTINUE
      ISHIFT = KREM + (ICYCLE-2)*N
C
C  ISHIFT IS A BIAS TO FIT THE EQUATIONS  INTO THE CORRECT ROWS
C  JSHIFT IS A BIAS TO FIT THE EQUATIONS  INTO THE CORRECT COLUMNS
C
  109 CONTINUE
C  ZERO OUT  KS22, KS21,  AS REQUIRED
      DO 8  I=1,K1
      G(I) = 0.
      DO 8 J =1,K2
      KS22(I,J) = 0.
    8 CONTINUE
      IPL = 1
      NLO = LIM1(ICYCLE)
      NHI = LIM2(ICYCLE)
      DO 100 J=NLO,NHI
      DO 100  I=1,6
      I2 = 1
      I1 = I
   15 IF(I1-3) 17, 18, 19
   19 I1 = I1-3
      I2 = I2+1
```

```
      GO TO 15
   18 IS = 3
      IT = I2
      IF ((P(L(J,3),I2)-PK).GT.0.) GO TO 99
C   HERE IF FORCE IS KNOWN
      P1 = P(L(J,3),I2)
      GO TO 25
   17 IF(I1-2) 20,21,21
   20 IS = 1
      IT = I2
      IF ((P(L(J,1),I2)-PK).GT.0.) GO TO 99
      P1 = P(L(J,1),I2)
      GO TO 25
   21 IS = 2
      IT = I2
      IF ((P(L(J,2),I2)-PK).GT.0.) GO TO 99
      P1 = P(L(J,2),I2)
   25 IF (2*(L(J,IS)-1)+IT-MSK(IPL)) 301,302,303
  302 III = IPL
      GO TO 327
  301 IMR = IPL - 1
  306 IF(2*(L(J,IS)-1) + IT-MSK(IMR)) 304, 305, 305
  305 III = IMR
      IPL = IMR
      GO TO 327
  304 IMR = IMR-1
      GO TO 306
  303 IMR = IPL + 1
  307 IF (2*(L(J,IS)-1)+IT-MSK(IMR)) 309, 309, 310
  309 III = IMR
      IPL = IMR
      GO TO 327
  310 IMR = IMR+1
      GO TO 307
  327 IF(III-ISHIFT.GT.K1)  GO TO 328
      IF(III-ISHIFT.LT.1)  GO TO 328
      IF(KS21(III-ISHIFT).NE.0.) GO TO  328
      KS21(III-ISHIFT) =KS21(III-ISHIFT) + P1
  328 CONTINUE
      DO 100  KJ = 1,2
      DO 100  KK = 1,3
      IF ((P(L(J,KK),KJ)-PK).GT.0.) GO TO 28
      IF (2*(L(J,KK)-1)+KJ-MSK(IPL)) 11, 12, 13
   12 II2 = IPL
      GO TO 27
   11 IMR = IPL-1
    6 IF(2*(L(J,KK)-1) + KJ - MSK(IMR)) 4,5,5
    5 II2 = IMR
      IPL = IMR
      GO TO 27
    4 IMR = IMR-1
      GO TO 6
   13 IMR = IPL + 1
    7 IF (2*(L(J,KK)-1) + KJ - MSK(IMR))  5, 5, 10
   10 IMR = IMR + 1
      GO TO 7
   27 IF(III-ISHIFT.GT.K1)  GO TO  99
      IF(III-ISHIFT.LT.1)  GO TO 99
      IF(II2-JSHIFT.GT.K2)  GO TO 999
      KS22(III-ISHIFT,II2-JSHIFT) = KS22(III-ISHIFT,II2-JSHIFT)
     1          +  KPR(I,3*(KJ-1)+KK,J )
      GO TO 99
```

```
   28 IF(DIL(J,KK),KJ).EQ.0.) GO TO 99
      IF(III-ISHIFT.GT.K1)  GO TO 99
      IF(III-ISHIFT.LT.1)  GO TO 99
      KS21(III-ISHIFT) = KS21(III-ISHIFT) -KPR(I,3*(KJ-1)+KK,J )
     1        * DIL(J,KK),KJ)
   99 CONTINUE
  100 CONTINUE
C   HAVE EQUATIONS, NOW SHIFT THE SUBMATRICES TO THE PROPER POSITION FOR
C   SUBROUTINE CHLSKY
      IF(ICYCLE.GT.1)  GO TO 430
C   S(1,1) IS (KREM X KREM), S(1,2) IS (KREM X N)
      DO 410  J = 1, KREM
      DO 410 I =  1,N
      LOCL = J + KREM*(I-1)
      LOCR1 = J +  32*(I-1+KREM)
C   SET UP S(1,2) IN S(1,1,4)
      Q(LOCL+3072) = Q(LOCR1)
  410 CONTINUE
      DO 420  J = 1, KREM
      DO 420 I = 1,KREM
      LOCL = J + KREM*(I-1)
      LOCR2 = J +   32*(I-1)
C   SET UP S(1,1) IN S(1,1,2)
      Q(LOCL+1024) = Q(LOCR2)
  420 CONTINUE
      GO TO 480
  430 CONTINUE
      IF (ICYCLE.GT.2)  GO TO 440
C   S(2,1) IS (N X KREM)   ,   S(2,2) AND S(2,3) ARE (N X N)
      K1 = KREM
      GO TO 450
  440 CONTINUE
C   S(I,I-1) , S(I,I) , S(I,I+1)  ARE (N X N)
      K1 = N
  450 CONTINUE
C   SET UP S(ICYCLE,ICYCLE+1) IN S(1,1,4)
      IF (ICYCLE.EQ.M) GO TO 461
      DO 460 I = 1,N
      DO 460 J = 1,N
      LOCL = J+N*(I-1)
      LOCR1 = J + 32*(I-1+K1+N)
      Q(LOCL+3072) = Q(LOCR1)
  460 CONTINUE
  461 CONTINUE
C   SET UP S(ICYCLE,ICYCLE) IN S(1,1,3)
      DO 468  I=1,N
      DO 468  J=1,N
      LOCR2 = J+32*(I-1+K1)
      LOCL = J + N * (I-1)
      Q(LOCL+2048) = Q(LOCR2)
  468 CONTINUE
C   MOVE  S(ICYCLE,ICYCLE) TO S(1,1,2)
      NSQ = N**2
      DO 470  II=1,NSQ
      Q(II+1024) = Q(II+2048)
  470 CONTINUE
C   REARRANGE S(ICYCLE,ICYCLE-1) WITHIN S(1,1,1)
      KLOC = 0
      DO 480 I=2,K1
      KLOC = KLOC+N
      DO 480 J = 1,N
      Q(KLOC+J) = S(J,I,1)
```

```
480 CONTINUE
    RETURN
999 CONTINUE
    PRINT 1000,ICYCLE
1000 FORMAT (H1,35H  BANDWIDTH EXCEEDED, PARTITION ROW,I3)
    IROW1 = II1 + ISHIFT
    JROW1 = II2 + JSHIFT
    PRINT 1001, IROW1, JROW1
1001 FORMAT(1H ,4HROW=,I3,6X,7HCOLUMN=,I3)
    PRINT 9120, II1, II2, ISHIFT, JSHIFT
9120 FORMAT(1H ,4HII1=,I3,4HII2=,I3,7HISHIFT=,I3,7HJSHIFT=,I3)
    PRINT 9121, K1, K2, KREM, N, M
9121 FORMAT(1H ,3HK1=,I3,3HK2=,I3,5HKREM=,I3,2HN=,I3,2HM=,I3)
    PRINT 9122,I, J, IPL, IMR, IS, IT
9122 FORMAT(1H ,2HI=,I3,2HJ=,I3,4HIPL=,I3,4HIMR=,I3,3HIS=,I3,3HIT=,I3)
    STOP
    END
```

```
      SUBROUTINE CHLSKY
C
C   THIS ROUTINE SOLVES   SU=G , WHERE S IS A TRI-DIAGONAL MATRIX IN
C   SUBMATRICES, WITH ELEMENTS OF ORDER N.
C   S IS KNOWN
C   SP IS A VECTOR OF DIMENSION (NXM) WHERE M IS THE NUMBER OF DIVISIONS OF S
C   C IS WRITTEN ONTO TAPE AFTER DERIVATION ON THE FORWARD PASS.
C   AND READ BACK IN ON THE BACKSWEEP
C   S(1,1,1)  INITIALLY CONTAINS  S(I,I-1)
C   S(1,1,2)  INITIALLY CONTAINS  S(I,I  )
C   S(1,1,4)  INITIALLY CONTAINS  S(I,I+1)
C   SP CORRESPONDS TO P IN THE WRITEUP BY GATEWOOD ON THE FORWARD PASS.
C   ON THE BACKSWEEP, IT CORRESPONDS TO U.
      COMMON /1/  S(32,32,5),L(300,3),G(32),SPACE(20)
     1          , NUF(3,314)        ,X(155),Y(155),CDA(3,6,250)
     2          , KPR(6,6,250),MSK(310),JTOTAL,N,KREM,M , IMR, IPL
     3          ,  P(155,2)   ,D(155,2)
      COMMON/LIM/  LIM1(10),LIM2(10)
      DIMENSION   SP(32,10)
      EQUIVALENCE  (NUF,SP)
      TYPE REAL KPR
      DIMENSION   C(1024)
      EQUIVALENCE  (S(4097),C)
      DATA (XLIMIT =1.E-8)
      REWIND 96
      N2 = 2*N
      KREM2 = 2*KREM
      DO 30 ICYCLE =1,M
      CALL BIGMX(ICYCLE)
      IF(ICYCLE-2) 1,2,3
    1 K1 = KREM
      K3 = KREM2
      GO TO 10
    2 K1 = N
      K2 = KREM
      K3 = N2
      GO TO 4
    3 K1 =N
      K2 =N
      K3 =N2
    4 CONTINUE
    5 IF (UNIT,96)  6,7 ,600, 600
    6 GO TO 5
    7 CONTINUE
      CALL MXMULT(S(1,1,1) , S(1,1,5) , S(1,1,3) ,K1,K2,K1)
C
C   S(1,1,5) CONTAINS C FROM LAST CYCLE
C
      CALL MXSUB(S(1,1,2) , S(1,1,3) , S(1,1,2) ,K1,K1)
   10 CONTINUE
C
C   B(I,I)  NOW IN S(1,1,2)
C
      CALL INVERT(S(1,1,2) , K1 , K3, XLIMIT , FLAG)
      IF (FLAG.NE.0.) GO TO 500
C
C   INVERSE OF B(I,I) NOW IN  S(1,1,2)
C
      IF (ICYCLE.EQ.1) GO TO 20
      CALL MXMULT(S(1,1,1) ,SP(1,ICYCLE-1) , S(1,1,3) , N, K2 , 1)
      CALL MXSUB (G  ,  S(1,1,3), G, N, 1)
   20 CONTINUE
```

```
      CALL MXMULT(S(1,1,2) , G ,SP(1, ICYCLE   ) , K1 , K1 ,  1)
      IF (ICYCLE.GE.M) GO TO 35
      CALL MXMULT (S(1,1,2) , S(1,1,4) ,.S(1,1,5), K1 , K1 , N)
      NSQ = K1*N
      BUFFER OUT  (96,1)  (C(1), C(NSQ ))
   30 CONTINUE
   35 CONTINUE
C
C   NOW IN BACKSWEEP, SOLVING FOR U
C
      DO 60 I = 2,M
      JCYCLE = M-I+1
      IF(JCYCLE.GT.1)  GO TO 36
      K1 = KREM
      GO TO 37
   36 K1 = N
   37 CONTINUE
      NSQ = K1*N
      IF (JCYCLE.EQ.M-1)  GO TO 41
      BACKSPACE 96
   41 CONTINUE
      BACKSPACE 96
      BUFFER IN    (96,1)  (C(1),C(NSQ ))
   42 IF(UNIT,96) 43, 44,700,700
   43 GO TO 42
   44 CONTINUE
C
C   U(M)=SP(M) , CONSIDER FIRST (M-1)TH CYCLE
C
      CALL MXMULT(S(1,1,5) ,SP(1,JCYCLE+1),S(1,1,1),K1,N, 1)
      CALL MXSUB(SP(1,JCYCLE), S(1,1,1) ,SP(1,JCYCLE), K1 ,  1)
   60 CONTINUE
C
C   U(NS,I) NOW STORED IN SP(N,I) , I=1,M
C
      RETURN
  500 CONTINUE
      PRINT 1000 , ICYCLE
 1000 FORMAT (31H1COULD NOT INVERT MATRIX IN ROW,I2)
      STOP
  600 CONTINUE
      PRINT 1001,ICYCLE
 1001 FORMAT(37H1         ERROR READING C INTO CORE ON I2, 7HTH ROW.)
      STOP
  700 CONTINUE
      PRINT 1002, JCYCLE
 1002 FORMAT(37H1         ERROR WRITING C ONTO TAPE ON I2, 7HTH ROW.)
      STOP
      END
```

```
      SUBROUTINE MXMULT(A,B,C,M,N,K)
C
C  THIS SUBROUTINE MULTIPLIES MATRIX A BY MATRIX B  AND STORES THE
C  PRODUCT IN C.    (C CANNOT BE THE SAME AS A OR B.)
C
C  A IS (M X N)
C  B IS (N X K)
C  C IS (M X K)
C
      DIMENSION  A(M,N) , B(N,K) , C(M,K)
C
      DO 1  I=1,M
      DO 1  L=1,K
      C(I,L) = 0.
      DO 1  J=1,N
      C(I,L) = C(I,L) + A(I,J) * B(J,L)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MXSUB(A,B,C,M,N)
C
C  THIS SUBROUTINE SUBTRACTS MATRIX B FROM MATRIX A,STORES RESULT IN C
C
C  A, B, AND C ARE (M X N)        (C CAN BE THE SAME AS A OR B)
C
      DIMENSION   A(M,N) , B(M,N) ,C(M,N)
C
      DO 1  I=1,M
      DO 1  J=1,N
      C(I,J) = A(I,J) - B(I,J)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MXCON(A,B,X,M,N)
C
C  THIS SUBROUTINE MULTIPLIES MATRIX A (MXN) BY CONSTANT X, RESULT IN B
C  A MAY BE SAME AS B.
C  THIS SUBROUTINE MULTIPLIES MATRIX A (MXN) BY CONSTANT X, RESULT IN B
C  A MAY BE SAME AS B.
      DIMENSION  A(M,N) , B(M,N)
      DO 1 I=1,M
      DO 1 J=1,N
      B(I,J)= X*A(I,J)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INVERT(B,K,K2,XMIN,FLAG)
C  THIS SUBROUTINE SETS UP A UNIT MATRIX ADJACENT TO B(K,K)
C  ELEMENTARY ROW OPERATIONS ARE THEN PERFORMED ON THE NEW K X 2K MATRIX
C  TO REDUCE B(K,K) TO A UNIT MATRIX.  THIS WILL PLACE THE INVERSE OF
C  THE MATRIX B(K,K) IN THE RIGHT HALF OF B(K,2K)
C  ON EXIT. THE INVERSE OF B REPLACES B
C   B9+S N ARRAY OF 2*K**2 LOCATIONS CONTAINING THE MATRIX
C  K IS THE DIMENSION OF THE SQUARE MATRIX B
C  K2 IS 2*K
C  XMIN IS THE SMALLEST ALLOWABLE MAGNITUDE OF THE PIVOT
C  FLAG WILL BE RETURNED AS 0. IF THE INVERSION WENT OFF OK
C  FLAG WILL BE RETURNED AS 10. IF A PIVOT ELEMENT WAS TOO SMALL
C  FLAG SHOULD BE TESTED AFTER EACH CALL TO THIS ROUTINE
C
      DIMENSION B(K,K2)
C
      FLAG = 0.
C
C  SET UP UNIT MATRIX
C
      IF(K.GT.1)  GO TO 20
      IF(ABS(B(1,1)).LT.XMIN)  GO TO 10
      B(1,1) = 1./B(1,1)
      RETURN
   20 CONTINUE
      DO 1  I=1,K
      DO 1  J=1,K
      B(I,K+J) = 0.
      IF(I.EQ.J)  B(I,K+J) = 1.
    1 CONTINUE
C
C  FIND LEADING ELEMENT WITH GREATEST MAGNITUDE
C
      DO 6  J=1,K
      M = J
      N = J+1
      IF(J.EQ.K) GO TO 21
      DO 2  L=N,K
      IF (ABS(B(M,J)).LT.ABS(B(L,J)))  M=L
    2 CONTINUE
   21 CONTINUE
      IF (ABS(B(M,J)).LT.XMIN)  GO TO 10
      IF(J.EQ.K) GO TO 31
C
C  INTERCHANGE JTH AND MTH ROWS
C
      DO 3  L=J,K2
      D = B(J,L)
      B(J,L) = B(M,L)
      B(M,L) = D
    3 CONTINUE
   31 CONTINUE
C
C  ZERO OUT PIVOTAL JTH COLUMN.  SKIPPING PIVOTAL JTH ELEMENT
C
C  DIVIDE JTH ROW BY PIVOT
C
      DO  4   M=N,K2
      B(J,M) = B(J,M) / B(J,J)
    4 CONTINUE
      DO 6  M=1,K
```

```
C
C   M DETERMINES ROW BEING MODIFIED. ONE WHOLE ROW AT A TIME
C
      IF ( M.EQ.J )   GO TO 6
      DO 5  L=N,K2
C
C   L DETERMINES ELEMENT IN THE MTH ROW
C
      B(M,L) = B(M,L) - B(M,J) * B(J,L)
    5 CONTINUE
    6 CONTINUE
C
C   INVERSE OF B IS NOW IN RIGHT HALF OF B(K,K2)
C   NOW MOVE B INVERSE TO WHERE B WAS
      DO 7  I=1,K
      DO 7  J=1,K
      B(I,J) = B(I,J+K)
    7 CONTINUE
      RETURN
   10 FLAG = 10.
      RETURN
      END
```

FUNDAMENTAL CASE THREE
(AXIAL LOADING)

```
      PROGRAM MAINA
C
C   THIS IS THE EXECUTIVE PROGRAM USED WITH SUBROUTINE BIGMX AND SUBROUTINE
C   CHLSKY  TO GENERATE AND SOLVE LARGE SYSTEMS OF LINEAR EQUATIONS
C
      REWIND 20                                                       CASE 3
      CALL TAPESKIP(20,6,0)
      CALL TAPESKIP(20,2,0)
    1 CONTINUE
      CALL INPUTA
C
C   NOW HAVE ALL K-PRIME AND CDA MATRICES
C
      CALL CHOLA
C
C   NOW HAVE SOLUTION U IN SP
C
      CALL STRESA
C
C   ALL STRESSES NOW PRINTED OUT
C
      GO TO 1
      END
                                                                      23 CARDS
```

```
      SUBROUTINE INPUTA
C
C   THIS SUBROUTINE READS AND PRINTS THE INPUTS FOR THE PLANAR FINITE
C  ELEMENT PROGRAM.  ALL INPUTS NOT READ ARE GENERATED HERE.
C
      COMMON /1/  S(32,32,5),L(160,3),G(32),SPACE(20),XNU(2),E(2)
     1           , NUF(3,160)          ,X(155),Y(155),CDA(3,6,160)
     2          , KPR(6,6,160),MSK(310),JTOTAL,N,KREM,M , IMR, IPL
     3           ,  P(155,2)   ,D(155,2) , EPT(3), PT(6,160), SIT(3,160)
      COMMON /STRSS/ ALF(155) , MSIZE , R , GAMMA
      EQUIVALENCE(XNU(1),XNUI),        (E(1),EI)
      COMMON/LIM/  LIM1(10),LIM2(10)
      DIMENSION   SP(32,10)
      DIMENSION NUYX(250),NUXY(250),  GXY(250), EX(250), EY(250)
      DIMENSION COMENT(10) ,          PTM(6)
      DIMENSION BUMP(10) , BUMP1(5)
      EQUIVALENCE  (NUF,SP)
      EQUIVALENCE  (C,CP)
      DIMENSION JL(79,3)
C
C  NUYX = NUXY, THIS MODIFICATION
      EQUIVALENCE  (NUXY,NUYX)
      DIMENSION  TH(155) , CP(3,3) , T1(3,3,2),T2(3,3),DTD(18),
     1  T(6,6),KMX(6,6), C(3,3),A(6,6,2),DO(3,6),DD(18),DTO(6,3)
      EQUIVALENCE  (S,F) , (S(901),VF), (S(1801),PHI),(S(2701),ER)
     1           , (S(3001),EF),(S(3901),NUR)
     2           ,  (DO,DD) , (DTO,DTD)  , (S(4356),TH)
     3        ,  (S(4511),CP) , (S(4521),T1), (S(4541),T2)
     4        ,  (S(4551),CABC) , (S(4581),T) , (S(4621),KMX)
     5     ,     (S(4671),A)
      DIMENSION EPT1(3,2)
      TYPE REAL NUXY
      TYPE INTEGER  BUMP, BUMP1
      TYPE REAL     NUYX, NUR, NUF, KPR, KMX
      DATA  (KTOTAL = 97)
      DATA(BUMP=  2,4,7,7,15,1 ,9 ,9 ,12,-1)
      DATA(BUMP1= 5,7,7,7,5 )   ,(PI= 3.1415927)
      DATA  (RAD = 57.29578)
      DATA  (XLIM=1.E-8)
      DATA (PK=999.)
      DATA ((DD(JJ),JJ=1,18) = 0., 0., 0., 1., 0., 0., 0., 0., 1.,
     1                         0., 0., 0., 0., 0., 1., 0., 1., 0. )
      DATA ((DTD(JJ),JJ=1,18) = 0., 1., 0., 0., 0., 0., 0., 0., 0.,
     1                          0., 0., 1., 0., 0., 1., 0., 1., 0.)
      DATA(((JL(I,J),I=1,79),J=1,3) =
     1    1,1,1,2,2,3,3,3,4,4,4,5,6,7,7,8,8,8,9,10,10,11,11,12,13,13,
     2    14,15,15,16,16,17,17,18,18,19,20,23,24,24,25,25,25,26,28,29,
     3    29,30,30,35,21,21,21,21,22,22,23,23,27,27,27,28,28,36,46,37,
     4    47,38,48,39,39,40,41,41,41,42,42,43,43,
     5    2,3,4,6,7,7,8,9,9,10,11,11,13,6,14,14,15,16,16,16,17,10,
     6    18,18,20,21,21,21,22,15,23,23,24,17,25,25,35,27,27,28,28,29,
     7    30,30,32,32,33,33,34,36,35,37,38,39,39,40,40,41,41,42,43,43,44,
     8    45,37,46,38,47,39,48,49,49,40,50,51,51,52,52,53,
     9    3,4,5,7,3,8,9,4,10,11,5,12,14,14,8,15,16,9,10,17,18,18,12,
     1    19,21,14,15,22,23,23,17,24,25,25,19,26,21,24,28,25,29,30,26,
     2    31,29,33,30,34,31,37,37,38,39,22,40,23,41,27,42,43,28,44,32,
     3    46,36,47,37,48,38,49,40,50,50,51,42,52,43,53,44 )
C
      READ 1000,(COMENT(I),I=1,10)
 1000 FORMAT(10A8)
      PRINT 1001,(COMENT(I),I=1,10)
 1001 FORMAT(1H1,10A8)
```

```
C
C   KTOTAL  - TOTAL NUMBER OF NODES CONSIDERED
C
C
C   READ AND PRINT
C
      JTOTAL = 158
      READ 1003,VF,GAMMA,E(1),XNU(1),BETA
      IF (VF.EQ.0.)  GO TO 95
      XNU(2) = XNU(1)/BETA
      E(2) = E(1)/GAMMA
      EPT1(1,1) = -XNU(1)
      EPT1(1,2) = -XNU(2)
      EPT1(2,1) = EPT1(1,1)
      EPT1(2,2) = EPT1(1,2)
      EPT1(3,1) = 0.
      EPT1(3,2) = 0.
      E(1) = E(1) /(1.-XNU(1)**2)
      E(2) = E(2) /(1.-XNU(2)**2)
      XNU(1) = XNU(1)/(1.-XNU(1))
      XNU(2) = XNU(2)/(1.-XNU(2))
 1003 FORMAT(5F10.4)
      PRINT 1018
      PRINT 1016
      PRINT 1017,  VF, GAMMA, EI, XNUI, BETA
 1016 FORMAT(1H ,13X,2HVF, 10X,5HGAMMA, 13X,2HEI, 11X,4HXNUI, 11X4HBETA)
 1017 FORMAT(1H ,5(E15.8))
 1018 FORMAT(////)

      S3 = SQRT(3.)
      S3O2 = S3/2.
C
      EPI = E(1)
      EPII=E(2)
C
      XNUPI = XNU(1)
      XNUPII= XNU(2)
      DO 200 I = 1,97
      TH(I) =1.
      ALF(I)=0.
      D(I,1) = 1000.
      D(I,2) = 1000.
      P(I,1) = 0.
      P(I,2) = 0.
  200 CONTINUE
      P(64,1) =1000.
      P(64,2) =1000.
      D(64,1) =0.
      D(64,2) =0.
      P(97,1) =1000.
      P(97,2) =1000.
      D(97,1) = 0.
      D(97,2) = 0.
      P(34,1) =1000.
      P(34,2) =1000.
      D(34,1) = 0.
      D(34,2) = 0.
      P( 1,1) =1000.
      P( 1,2) =1000.
      D( 1,1) = 0.
      D( 1,2) = 0.
      I = 0
```

```
      DO 201   J= 1,10
      I = I +BUMP(J)
      D(I,2) =0.
      P(I,2) =1000.
      D(98-I,2) =0.
      P(98-I,2) =1000.
  201 CONTINUE
      I = 0
      DO  202 J = 1,5
      I = I+ BUMP1(J)
      D(I,1) =0.
      D(98-I,1) =0.
      P(I,1) = 1000.
      P(98-I,1) = 1000.
  202 CONTINUE
      R= SQRT(2.*S3*VF/PI)
      X(1) = S302
      Y(1) = .5
      DO 210 I=1,4
      X(I+1) = S302 - R/4.*COS(PI*(I-1)/6.)
      Y(I+1) = .5 -  R/4. *SIN(PI*(I-1)/6.)
  210 CONTINUE
C
      DO 220  I=1,7
      X(I+5) = S302 - R/2.* COS(PI*(I-1)/12.)
      Y(I+5) = .5   - R/2.* SIN(PI*(I-1)/12.)
C
      X(I+12) =  S302 - 3.*R/4.*COS(PI*(I-1)/12.)
      Y(I+12) =  .5  -  3.*R/4.*SIN(PI*(I-1)/12.)
C
      X(I+19) =  S302 - R * COS(PI*(I-1)/12.)
      Y(I+19) =  .5  -  R * SIN(PI*(I-1)/12.)
  220 CONTINUE
      X(34) = S302
      Y(34) = - .5
      X(31) = S302
      Y(31) = (Y(26)+Y(34)) / 2.
      X(45) = -.5* TAN(PI/6.)
      Y(45) = .5
      DX = (1.-R)/(2.*COS(PI/6.))
      X(36) = X(45) + DX
      Y(36) = .5
      X(35) = (X(20)+X(36))/2.
      Y(35) = .5
C
      DO 230 I = 1,4
      X(I+45) = (4 -I)* X(45) /4.
      Y(I+45) = (4 -I)* Y(45) /4.
  230 CONTINUE
C
      DELX = X(46) -X(45)
      DELY = Y(46)-Y(45)
      DO 240 I = 1,8
      X(I+36) = X(I+35) +DELX
      Y(I+36) = Y(I+35) +DELY
  240 CONTINUE
      X(32) = X(44) +(X(34)-X(44))/3.
      Y(32) = -.5
      X(33) = 2.*X(32) - X(44)
      Y(33) = -.5
      X(27) = X(23) + (X(32)-X(23))/3.
      Y(27) = Y(23) + (Y(32)-Y(23))/3.
```

```
      X(28) = 2.*X(27)-X(23)
      Y(28) = 2.*Y(27)-Y(23)
      X(29) = X(28) +(X(31)-X(28))/3.
      Y(29) = Y(28) +(Y(31)-Y(28))/3.
      X(30) = 2.*X(29)-X(28)
      Y(30) = 2.*Y(29)-Y(28)
C
C
      DO 250 I=50,97
      X(I) = -X(98-I)
      Y(I) = -Y(98-I)
  250 CONTINUE
C
C
C   PRINT OUT NODAL DATA
      I = 1
   12 CONTINUE
      LINE = 4
      PRINT 1010
   13 CONTINUE
      PRINT 1011, I, X(I), Y(I), P(I,1), P(I,2), D(I,1), D(I,2),
     1             TH(I), ALF(I)
      I = I+1
      IF(I.GT.KTOTAL)  GO TO 14
      LINE =  LINE + 2
      IF(LINE.GT.56) GO TO 12
      GO TO 13
   14 CONTINUE
 1011 FORMAT(1H  I4,7(3XE11.4),5X,F11.2/)
 1010 FORMAT(5H1NODE,7X,1HX,13X,1HY,13X,2HP1,12X,2HP2,12X,2HD1,12X,2HD2,
     1          8X,9HTHICKNESS,9X,5HALPHA /)
C
C   I       - NUMBER OF NODE
C   X       - X- COORDINATE OF ITH NODE
C   Y       - Y- COORDINATE OF ITH NODE
C   P(I,1) - KNOWN AND UNKNOWN FORCE COMPONENTS ALONG 1 DIRECTION
C   P(I,2) - KNOWN AND UNKNOWN FORCE COMPONENTS ALONG 2 DIRECTION
C   D(I,1) - KNOWN AND UNKNOWN DISPLACEMENTS ALONG 1 DIRECTION
C   D(I,2) - KNOWN AND UNKNOWN DISPLACEMENTS ALONG 2 DIRECTION
C   ALF(I) - ANGLE BETWEEN X DIRECTION AND 1 DIRECTION (POSITIVE WHEN
C             COUNTER-CLOCKWISE)
C   TH(I)  - PLATE THICKNESS AT ITH NODE
C
C   GET MSK MATRIX
      JP = 0
      DO 27  J=1,KTOTAL
      DO 27  I=1, 2
      IF (P(J,I).GT.PK)  GO TO 27
      JP = JP+1
      MSK(JP) = 2 * (J-1) + I
   27 CONTINUE
C    MSK  IS MATRIX OF INDICES OF KNOWN FORCES
C    IF FORCE P IS UNKNOWN, IT IS INPUT AS 1000.
C   NOW READ IN TRIANGLE DATA
C
C   JTOTAL - TOTAL NUMBER OF TRIANGLES
      DO 19 I=1,79
      DO 19 J=1,3
   19 L(I,J) = JL(I,J)
      DO 260 I= 80,158
      DO 260 J= 1,3
      L(I,J) = 98 - L(159-I,J)
```

```
  260 CONTINUE
C   L(J,1) - INDEX OF THE FIRST  NODE OF THE JTH TRIANGLE
C   L(J,2) - INDEX OF THE SECOND NODE OF THE JTH TRIANGLE
C   L(J,3) - INDEX OF THE THIRD  NODE OF THE JTH TRIANGLE
C
      DO 20 I=1,JTOTAL
      IF(I.LE.36.OR.I.GE.123) GO TO 300
      EX(I) = EPII
      EY(I) = EPII
      NUXY(I) =XNUPII
      GXY(I) = EPII/(2.*(1.+XNUPII))
      GO TO 310
  300 EX(I) = EPI
      EY(I) = EPI
      NUXY(I) = XNUPI
      GXY(I) = EPI  / (2.*(1.+XNUPI))
  310 CONTINUE
   20 CONTINUE
C   PRINT OUT TRIANGLE DATA
      LINE = 4
      PRINT  1012
      DO 24 I=1,JTOTAL
      IF(LINE.LT.54)  GO TO 22
      LINE = 4
      PRINT 1012
 1012 FORMAT(1H1,8HTRIANGLE,4X6HNODE 1,3X,6HNODE 2,3X,6HNODE 3,18X,2HEX,
     1    18X 2HEY,16X,4HNUYX,17X,3HGXY//)
   22 LINE = LINE+2
   24 PRINT 1023, I, (L(I,J),J=1,3),EX(I),EY(I),NUYX(I),GXY(I)
 1023 FORMAT(1H ,5X,I3,7X,I3,2(6X,I3),4(5X,E15.8)/)
C
C
      MSIZE = 156
      KREM = 6
      N = 25
      M = MSIZE / N
      IF(KREM.NE.N)   M = M+1
C
      LIM1(1) =    1
      LIM1(2) =    4
      LIM1(3) =    25
      LIM1(4) =    50
      LIM1(5) =    74
      LIM1(6) =    96
      LIM1(7) =    128
      LIM2(7) =    158
      LIM2(6) =    139
      LIM2(5) =    122
      LIM2(4) =    91
      LIM2(3) =    67
      LIM2(2) =    37
      LIM2(1) =    12

C
C  PRINT OUT PARTITION INFORMATION
      PRINT 1008
 1008 FORMAT(1H1,12X,20HTRIANGLES CONSIDERED)
      DO 31  I=1,M
      ISIZE = N
      IF(I.EQ.1)  ISIZE = KREM
      PRINT 1009,  I,LIM1(I),LIM2(I),  ISIZE
   31 CONTINUE
```

```
 1009 FORMAT(10H PARTITION,6X 5HFIRST,2X,2HTO,2X,4HLAST, 6X,9HDIMENSION/
     1    4X,I3,11X,I3,6X,I3,10X,I3)
      DO 28 I = 1,KTOTAL
      ALF(I) = ALF(I)/RAD
   28 CONTINUE
C   ALL ANGLES NOW IN RADIANS
      DO 400 I=1, JTOTAL
C   I   IS A TRIANGLE COUNTER
      KK3 = 2
      IF(I.LE.36.OR.I.GE.123) KK3 = 1
      B11 = 1./EX(I)
      B12 = -NUYX(I) / EY(I)
      B22 = 1. / EY(I)
      B33 = 1. / GXY(I)
      DELTA = B11*B22 -B12**2
      CP(1,1)   = B22 / DELTA
      CP(2,1)   = - B12 / DELTA
      CP(3,1)   =  0.
      CP(1,2) = CP(2,1)
      CP(2,2)   = B11/ DELTA
      CP(3,2)   =  0.
      CP(1,3)   =  0.
      CP(2,3)   =  0.
      CP(3,3) = 1./B33
C
C   C NOW IN C(3,3)   , MATRIX I
C
   30 CONTINUE
      THOMEG   = X(L(I,2))*Y(L(I,3)) +X(L(I,1))*Y(L(I,2))
     1         + Y(L(I,1))*X(L(I,3)) -X(L(I,3))*Y(L(I,2))
     2         - X(L(I,1))*Y(L(I,3)) -Y(L(I,1))*X(L(I,2))
      THOMEG=(TH(L(I,1))+TH(L(I,2))+TH(L(I,3)))/ 6. * THOMEG
      XI2 = X(L(I,2))- X(L(I,1))
      XI3 = X(L(I,3))- X(L(I,1))
      ETA2= Y(L(I,2))- Y(L(I,1))
      ETA3= Y(L(I,3))- Y(L(I,1))
      DELTA = XI2*ETA3 - XI3*ETA2
      DO 38 II = 1,3
      DO 38 JJ = 1,3
      A(II+3,JJ,1) = 0.
      A(II,JJ+3,1) = 0.
      A(II+3,JJ,2) = 0.
      A(II,JJ+3,2) = 0.
   38 CONTINUE
      A(1,1,1) = 1.
      A(2,1,1) = -(ETA3-ETA2) / DELTA
      A(3,1,1) = (XI3 - XI2) / DELTA
      A(1,2,1) = 0.
      A(2,2,1) = ETA3 / DELTA
      A(3,2,1) = -XI3 / DELTA
      A(1,3,1) = 0.
      A(2,3,1) = -ETA2 / DELTA
      A(3,3,1) = XI2 / DELTA
      DO 39 II = 1,3
      DO 39 JJ = 1,3
      A(II+3,JJ+3,1) = A(II,JJ,1)
      A(II,JJ,2) = A(JJ,II,1)
      A(II+3,JJ+3,2) = A(JJ,II,1)
   39 CONTINUE
C
C   TRANSPOSE OF INVERSE OF A NOW IN A(1,1,2) , A INVERSE STILL IN A
C
```

```
      CALL MXMULT(D0,A,KMX(1,1),3,6,6)
      CALL MXMULT(C,FPT1(1,KK3),SIT(1,I),3,3,1 )
      CALL MXMULT(C,KMX(1,1)   ,CDA(1,1,I),3,3,6)
C   PRODUCT C*D*(A**-1) NOW IN CDA(1,1,I), ITH TRIANGLE
C
      CALL MXMULT(A(1,1,2),DT0,A,6,6,3)
      CALL MXMULT(EPT1(1,KK3),CDA(1,1,I),PTM,1,3,6)
      CALL MXCON(PTM,PTM,THOMFG,1,6)
      CALL MXCON(A,KPR(1,1,I),THOMFG,6,3)
      CALL MXMULT(KPR(1,1,I),CDA(1,1,I),KMX,6,3,6)
C
C  MATRIX K(I) NOW IN KMX, TRIANGLE  I
      DO 40 II =1,6
      DO 40 JJ =1,6
      T(II,JJ)    = 0.
   40 CONTINUE
      T(1,1)    =  COS(ALF(L(I,1)))
      T(4,1)    =  SIN(ALF(L(I,1)))
      T(2,2)    =  COS(ALF(L(I,2)))
      T(5,2)    =  SIN(ALF(L(I,2)))
      T(3,3)    =  COS(ALF(L(I,3)))
      T(6,3)    =  SIN(ALF(L(I,3)))
      T(1,4)    = -T(4,1)
      T(4,4)    =  T(1,1)
      T(2,5)    = -T(5,2)
      T(5,5)    =  T(2,2)
      T(3,6)    = -T(6,3)
      T(6,6)    =  T(3,3)
      CALL MXMULT (KMX,T,A,6,6,6)
      CALL MXMULT(T,PTM,PT(1,I),6,6,1)
      T(4,1) = - T(4,1)
      T(5,2) = - T(5,2)
      T(6,3) = - T(6,3)
      T(1,4) = - T(1,4)
      T(2,5) = - T(2,5)
      T(3,6) = - T(3,6)
C
C  INVERSE OF T NOW IN T
C
      CALL MXMULT (T,A,KPR(1,1,I),6,6,6)
C
C  K-PRIME NOW IN KPR , A  HAS BEEN CLOBBERED.
  400 CONTINUE
      RETURN
  800 PRINT 1051,II,JJ,II
 1051 FORMAT(1H1, 5H  EF(,I3,1H,I3,7H) = ER(,I3,6H) = 0.)
      STOP
  900 CONTINUE
      PRINT 1050    , J
 1050 FORMAT(1H1 35HCOULD NOT INVERT MATRIX  T1,TRIANGLE,I3)
  950 CONTINUE
      REWIND 20                                                  CASE 3
      STOP
      END
                                                              1406 CARDS
```

```
      SUBROUTINE STRESA
C
C  THIS SUBROUTINE DERIVES AND PRINTS STRESSES
C
      COMMON /1/  S(32,32,5),L(160,3),G(32),SPACE(20),XNU(2),E(2)
     1            , NUF(3,160)        ,X(155),Y(155),CDA(3,6,160)
     2           , KPR(6,6,160),MSK(310),JTOTAL,N,KREM,M , IMR, IPL
     3           ,  P(155,2)   ,D(155,2)  , EPT(3), PT(6,160), SIT(3,160)
      COMMON /STRSS/ ALF(155) , MSIZE , R , GAMMA
      EQUIVALENCE(XNU(1),XNU1),        (E(1),E1)
      COMMON/LIM/  LIM1(10),LIM2(10)
      DIMENSION  DVX(6), SIGOUT(632)
      EQUIVALENCE (S,SIGOUT)
      DIMENSION   SP(32,10)
      EQUIVALENCE  (NUF,SP)
      TYPE REAL KPR
      EQUIVALENCE  (S,SIG) , (S(931),PSTR) , (S(1861),X0),(S(2161),Y0)
      EQUIVALENCE  (S(2461),DEL) , (S(2761),DX)
      DIMENSION   ERR(310)
      DIMENSION   DX(6) , SIG(4,310)  ,   PSTR(3,310)
      DIMENSION  X0(300) , Y0(300), DEL(310)
      DIMENSION  KS22(32,96) ,PZ(160)
      EQUIVALENCE   (KPR,KS22)
      TYPE REAL KS22
      DATA (PK=999.)
      E(1) = E(1)*(1.-(XNU(1)/(1.+XNU(1))))**2)
      E(2) = E(2)*(1.-(XNU(2)/(1.+XNU(2))))**2)
C  REMOVE GAPS FROM SP(32,10) = DEL(310)
      DO 5 J=1,KREM
      DEL(J) = SP(J,1)
    5 CONTINUE
      KLOC = KREM -N
      DO 10 I = 2,M
      KLOC = KLOC + N
      DO 10 J = 1,N
      DEL(KLOC+J) = SP(J,I)
   10 CONTINUE
      PRINT 1010
 1010 FORMAT(1H1,50X,13HDISPLACEMENTS,//)
      NDEL = MSIZE/7
      JCNT = 0
      DO 15 J=1,NDEL
      JCNT = JCNT + 1
      IF(JCNT.GT.18)  GO TO 14
      JFIR = 7*(J-1) + 1
      JLAST =  JFIR + 6
      PRINT 1011 , (K,K=JFIR,JLAST)
 1011 FORMAT(1H ,7(8X,4HDEL(,I3,1H) ))
      PRINT 1012, (DEL(K),K=JFIR,JLAST)
 1012 FORMAT(1H ,7(2X,E14.7)/)
      GO TO 15
   14 PRINT 1010
      JCNT = 0
   15 CONTINUE
      LOC1 = 7*NDEL+1
      LOC2 = MSIZE
      IF(LOC1.GT.LOC2)  GO TO 20
      PRINT 1011,(K,K=LOC1,LOC2)
      PRINT 1012, (DEL(K),K=LOC1,LOC2)
   20 CONTINUE
      DO 855  I=1,JTOTAL
      J = I
```

```
      KK3 = 2
      IF(I.LE.36.OR.I.GE.123) KK3 = 1
      KZ = 0
      DO 831 KK =1,3
      DO 832 KJ =1,2
      KZ = KZ + 1
      IF( P(L(J,KK),KJ).GT.PK) GO TO 828
      IF(2*(L(J,KK)-1)+KJ-MSK(IPL)) 801, 802, 803
  802 II2 = IPL
      GO TO 827
  801 IMR = IPL - 1
  806 IF(2*(L(J,KK)-1) +KJ -MSK(IMR))  804, 805, 805
  805 II2 = IMR
      IPL = IMR
      GO TO 827
  804 IMR = IMR - 1
      GO TO 806
  803 IMR = IPL + 1
  807 IF(2*(L(J,KK)-1) +KJ -MSK(IMR)) 805, 805, 810
  810 IMR = IMR + 1
      GO TO 807
  827 DVX(KZ) = DEL(II2)
      GO TO 832
  828 DVX(KZ)= D(L(J,KK),KJ)
  832 CONTINUE
      DX(KZ-1) = DVX(KZ-1)*COS(ALF(L(J,KK)))-DVX(KZ)*SIN(ALF(L(J,KK)))
      DX(KZ) = DVX(KZ-1)*SIN(ALF(L(J,KK)))+DVX(KZ)*COS(ALF(L(J,KK)))
  831 CONTINUE
      DX2 = DX(2)
      DX(2) = DX(3)
      DX(3) = DX(5)
      DX(5) = DX(4)
      DX(4) = DX2
      CALL MXMULT(CDA(1,1,I),DX, SIG(1,I),3,6,1)
      DO 838 J=1,3
  838 SIG(J,I) = SIG(J,I) - SIT(J,I)
      SIG(4,I) = SIG(3,I)
      Y1 = Y(L(I,1))
      Y2 = Y(L(I,2))
      Y3 = Y(L(I,3))
      AREA = (X(L(I,1))*(Y2-Y3)+X(L(I,2))*(Y3-Y1)+X(L(I,3))*(Y1-Y2))/2.
      SIGZ = (SIG(1,I)+SIG(2,I))*XNU(KK3)/(1.+XNU(KK3))+E(KK3)
      SIG(3,I) = SIGZ
      PZ(I) = SIGZ*AREA
C  SIGMA NOW IN SIG(1,I)   , TRIANGLE I
      XO(I) = 0.
      YO(I) = 0.
      DO 840  K = 1,3
      XO(I) = XO(I) + X(L(I,K))
      YO(I) = YO(I) + Y(L(I,K))
  840 CONTINUE
      XO(I) = XO(I) / 3.
      YO(I) = YO(I) / 3.
  850 CONTINUE
  855 CONTINUE
      ENBARZ = 0.
      DO 860 I=1,158
  860 ENBARZ = ENBARZ + PZ(I)
      ENBARZ = SQRT(3.)/ENBARZ
      DO 865 I=1,632
  865 SIGOUT(I) = SIGOUT(I)*ENBARZ
      I = 1
```

```
 870 CONTINUE
     PRINT 1000
     LINE = 3
1000 FORMAT(1H1,2(8HTRIANGLE,6X,8HCENTROID,9X,12HVECTOR SIGMA,10X)/)
 871 CONTINUE
     II = I+1
     DO  880  J=1,4
     GO TO (873,874,875,875), J
 873 PRINT 1001, I, XO(I), SIG(J,I),II,XO(II), SIG(J,II)
     GO TO 878
 874 PRINT 1002,    YO(I),SIG(J,I),     YO(II), SIG(J,II)
     GO TO 878
 875 PRINT 1003, SIG(J,I), SIG(J,II)
 878 CONTINUE
 880 CONTINUE
     LINE = LINE + 6
     I = I+ 2
     PRINT 1004
1001 FORMAT(1H 2(5X,I3,F14.6, 7X,E15.8,10X))
1002 FORMAT(1H 2(8X,F14.6, 7X,E15.8,10X) )
1003 FORMAT(1H 2(29X,E15.8,10X))
1004 FORMAT(/)
     IF(I.GT.JTOTAL)  GO TO 890
     IF (LINE.GT.54) GO TO 870
     GO TO 871
 890 CONTINUE
     WRITE (20)  (SIGOUT(I),I=1,316)                              CASE 3
     SIGMA = 0.
     DO 900 I=1,JTOTAL
     SIGMA = SIGMA+PZ(I)
 900 CONTINUE
     EC = SIGMA/SQRT(3.)
     CHI = EC/E(1)
     PRINT 1017 , CHI,EC
1017 FORMAT( 7H1  CHI=, E13.6,4X,6H    EC=,E13.6)
     PRINT 8787, ENBARZ
8787 FORMAT(//23H NORMALIZATION FACTOR =,2X,E15.7)
     RETURN
     END
```

```
      SUBROUTINE CHOLA
C
C  THIS ROUTINE SOLVES  SU=G , WHERE S IS A TRI-DIAGONAL MATRIX IN
C  SUBMATRICES, WITH ELEMENTS OF ORDER N.
C  S IS KNOWN
C  SP IS A VECTOR OF DIMENSION (NXM) WHERE M IS THE NUMBER OF DIVISIONS OF S
C  C IS WRITTEN ONTO TAPE AFTER DERIVATION ON THE FORWARD PASS.
C  AND READ BACK IN ON THE BACKSWEEP
C  S(1,1,1)  INITIALLY CONTAINS  S(I,I-1)
C  S(1,1,2)  INITIALLY CONTAINS  S(I,I  )
C  S(1,1,4)  INITIALLY CONTAINS  S(I,I+1)
C  SP CORRESPONDS TO P IN THE WRITEUP BY GATEWOOD ON THE FORWARD PASS.
C  ON THE BACKSWEEP, IT CORRESPONDS TO U.
      COMMON /1/  S(32,32,5),L(160,3),G(32),SPACE(20),XNU(2),E(2)
     1          , NUF(3,160)        ,X(155),Y(155),CDA(3,6,160)
     2          , KPR(6,6,160),MSK(310),JTOTAL,N,KREM,M , IMP, IPL
     3          ,  P(155,2)  ,D(155,2) , EPT(3), PT(6,160), SIT(3,160)
      COMMON/LIM/  LIM1(10),LIM2(10)
      DIMENSION   SP(32,10)
      EQUIVALENCE  (NUF,SP)
      TYPE REAL KPR
      DIMENSION   C(1024)
      EQUIVALENCE  (S(4097),C)
      DATA (XLIMIT =1.E-8)
      REWIND 96
      N2 = 2*N
      KREM2 = 2*KREM
      DO 30 ICYCLE =1,M
      CALL BIGMXA(ICYCLE)
      IF(ICYCLE-2) 1,2,3
    1 K1 = KREM
      K3 = KREM2
      GO TO 10
    2 K1 = N
      K2 = KREM
      K3 = N2
      GO TO 4
    3 K1 =N
      K2 =N
      K3 =N2
    4 CONTINUE
    5 IF (UNIT,96)  6,7 ,600, 600
    6 GO TO 5
    7 CONTINUE
      CALL MXMULT(S(1,1,1) , S(1,1,5) , S(1,1,3) ,K1,K2,K1)
C
C  S(1,1,5) CONTAINS C FROM LAST CYCLE
C
      CALL MXSUB(S(1,1,2) , S(1,1,3) , S(1,1,2) ,K1,K1)
   10 CONTINUE
C
C  B(I,I)  NOW IN S(1,1,2)
C
      CALL INVERT(S(1,1,2) , K1 , K3, XLIMIT , FLAG)
      IF (FLAG.NE.0.) GO TO 500
C
C  INVERSE OF B(I,I) NOW IN  S(1,1,2)
C
      IF (ICYCLE.EQ.1) GO TO 20
      CALL MXMULT(S(1,1,1) ,SP(1,ICYCLE-1) , S(1,1,3) , N, K2 , 1)
      CALL MXSUB (G  ,  S(1,1,3), G, N, 1)
   20 CONTINUE
```

```
      CALL MXMULT(S(1,1,2) , G ,SP(1, ICYCLE   ) , K1 , K1 ,  1)
      IF (ICYCLE.GE.M) GO TO 35
      CALL MXMULT (S(1,1,2) , S(1,1,4) , S(1,1,5), K1 , K1 , N)
      NSQ = K1*N
      BUFFER OUT  (96,1)  (C(1), C(NSQ ))
   30 CONTINUE
   35 CONTINUE
C
C   NOW IN BACKSWEEP, SOLVING FOR U
C
      DO 60 I = 2,M
      JCYCLE = M-I+1
      IF(JCYCLE.GT.1)  GO TO 36
      K1 = KREM
      GO TO 37
   36 K1 = N
   37 CONTINUE
      NSQ = K1*N
      IF (JCYCLE.EQ.M-1)  GO TO 41
      BACKSPACE 96
   41 CONTINUE
      BACKSPACE 96
      BUFFER IN   (96,1)  (C(1),C(NSQ ))
   42 IF(UNIT,96) 43, 44,700,700
   43 GO TO 42
   44 CONTINUE
C
C   U(M)=SP(M) , CONSIDER FIRST (M-1)TH CYCLE
C
      CALL MXMULT(S(1,1,5) ,SP(1,JCYCLE+1),S(1,1,1),K1,N, 1)
      CALL MXSUB(SP(1,JCYCLE), S(1,1,1) ,SP(1,JCYCLE), K1 ,  1)
   60 CONTINUE
C
C   U(NS,1) NOW STORED IN SP(N,I) , I=1,M
C
      RETURN
  500 CONTINUE
      PRINT 1000 , ICYCLE
 1000 FORMAT (31HICOULD NOT  INVERT MATRIX IN ROW,I2)
      STOP
  600 CONTINUE
      PRINT 1001,ICYCLE
 1001 FORMAT(37H1        ERROR READING C INTO CORE ON I2, 7HTH ROW.)
      STOP
  700 CONTINUE
      PRINT 1002, JCYCLE
 1002 FORMAT(37H1        ERROR WRITING C ONTO TAPE ON I2, 7HTH ROW.)
      STOP
      END
```

```
      SUBROUTINE BIGMXA(ICYCLE)
C
C   THIS SUBROUTINE SEARCHES THE K-PRIME MATRICES BETWEEN LIM1(ICYCLE)
C   AND LIM2(ICYCLE) TO SET UP THE KREM OR N  ROWS OF S=K*22 REQUIRED
C   BY THE EQUATION SOLVING ROUTINE CHLSKY
C
      COMMON /1/  S(32,32,5),L(160,3),G(32),SPACE(20),XNU(2),E(2)
     1          , NUF(3,160)          ,X(155),Y(155),CDA(3,6,160)
     2          , KPR(6,6,160),MSK(310),JTOTAL,N,KREM,M , IMR, IPL
     3          ,  P(155,2)  ,D(155,2) , EPT(3), PT(6,160), SII(3,160)
      COMMON/LIM/  LIM1(10),LIM2(10)
      DIMENSION   SP(32,10)
      EQUIVALENCE  (NUF,SP)
      DIMENSION  KS22(32,96)     ,  KS21(32)
      DIMENSION  Q(5120)
      EQUIVALENCE  (S,KS22,Q)   , (G,KS21)
      TYPE REAL KPR
      TYPE REAL  KS22 , KS21
      DATA  (PK = 999.)
      IF(ICYCLE.GT.1) GO TO 1
      ISHIFT =0
      JSHIFT =0
      K1 = KREM
      K2 = KREM + N
      GO TO 109
    1 CONTINUE
      IF (ICYCLE.EQ.M)  GO TO 3
      K1 = N
      K2 = 3*N
      GO TO  2
    3 CONTINUE
      K1 = N
      K2 = 2*N
    2 CONTINUE
      IF(ICYCLE.GT.2)  GO TO 110
      JSHIFT = 0
      GO TO 111
  110 CONTINUE
      JSHIFT = KREM + (ICYCLE-3)*N
  111 CONTINUE
      ISHIFT = KREM + (ICYCLE-2)*N
C
C  ISHIFT IS A BIAS TO FIT THE EQUATIONS  INTO THE CORRECT ROWS
C  JSHIFT IS A BIAS TO FIT THE EQUATIONS  INTO THE CORRECT COLUMNS
C
  109 CONTINUE
C  ZERO OUT  KS22, KS21,  AS REQUIRED
      DO 8  I=1,K1
      G(I) = 0.
      DO 8 J =1,K2
      KS22(I,J) = 0.
    8 CONTINUE
      IPL = 1
      NLO = LIM1(ICYCLE)
      NHI = LIM2(ICYCLE)
      DO 100 J=NLO,NHI
      DO 100  I=1,6
      I2 = 1
      I1 = I
   15 IF(I1-3) 17, 18, 19
   19 I1 = I1-3
      I2 = I2+1
```

```
      GO TO 15
   18 IS = 3
      IT = I2
      IF ((P(L(J,3),I2)-PK).GT.0.) GO TO 99
C   HERE IF FORCE IS KNOWN
      P1 = P(L(J,3),I2)
      GO TO 25
   17 IF(I1-2) 20,21,21
   20 IS = 1
      IT = I2
      IF ((P(L(J,1),I2)-PK).GT.0.) GO TO 99
      P1 = P(L(J,1),I2)
      GO TO 25
   21 IS = 2
      IT = I2
      IF ((P(L(J,2),I2)-PK).GT.0.) GO TO 99
      P1 = P(L(J,2),I2)
   25 CONTINUE
      IF (2*(L(J,IS)-1)+IT-MSK(IPL)) 301,302,303
  302 III = IPL
      GO TO 327
  301 IMR = IPL - 1
  306 IF(2*(L(J,IS)-1) + IT-MSK(IMR)) 304, 305, 305
  305 III = IMR
      IPL = IMR
      GO TO 327
  304 IMR = IMR-1
      GO TO 306
  303 IMR = IPL + 1
  307 IF (2*(L(J,IS)-1)+IT-MSK(IMR)) 309, 309, 310
  309 III = IMR
      IPL = IMR
      GO TO 327
  310 IMR = IMR+1
      GO TO 307
  327 IF(III-ISHIFT.GT.K1)  GO TO 328
      IF(III-ISHIFT.LT.1)  GO TO 328
      KS21(III-ISHIFT) =KS21(III-ISHIFT) + PT(I,J)
  328 CONTINUE
      DO 100  KJ = 1,2
      DO 100  KK = 1,3
      IF ((P(L(J,KK),KJ)-PK).GT.0.) GO TO 28
      IF (2*(L(J,KK)-1)+KJ-MSK(IPL)) 11, 12, 13
   12 II2 = IPL
      GO TO 27
   11 IMR = IPL-1
    6 IF(2*(L(J,KK)-1) + KJ - MSK(IMR)) 4,5,5
    5 II2 = IMR
      IPL = IMR
      GO TO 27
    4 IMR = IMR-1
      GO TO 6
   13 IMR = IPL + 1
    7 IF (2*(L(J,KK)-1) + KJ - MSK(IMR))  5, 5, 10
   10 IMR = IMR + 1
      GO TO 7
   27 IF(III-ISHIFT.GT.K1)  GO TO  99
      IF(III-ISHIFT.LT.1)  GO TO 99
      IF(II2-JSHIFT.GT.K2)  GO TO 999
      KS22(III-ISHIFT,II2-JSHIFT) = KS22(III-ISHIFT,II2-JSHIFT)
     1          + KPR(I,3*(KJ-1)+KK,J )
      GO TO 99
```

```
   28 IF(D(L(J,KK),KJ).EQ.0.) GO TO 99
      IF(III-ISHIFT.GT.K1)  GO TO 99
      IF(III-ISHIFT.LT.1)  GO TO 99
      KS21(III-ISHIFT) = KS21(III-ISHIFT) -KPR(I,3*(KJ-1)+KK,J )
     1        * D(L(J,KK),KJ)
   99 CONTINUE
  100 CONTINUE
C  HAVE EQUATIONS, NOW SHIFT THE SUBMATRICES TO THE PROPER POSITION FOR
C  SUBROUTINE CHLSKY
      IF(ICYCLE.GT.1)  GO TO 430
C  S(1,1) IS (KREM X KREM), S(1,2) IS (KREM X N)
      DO 410  J = 1, KREM
      DO 410 I =  1,N
      LOCL = J + KREM*(I-1)
      LOCR1 = J +   32*(I-1+KREM)
C  SET UP S(1,2) IN S(1,1,4)
      Q(LOCL+3072) = Q(LOCR1)
  410 CONTINUE
      DO 420  J = 1, KREM
      DO 420 I = 1,KREM
      LOCL = J + KREM*(I-1)
      LOCR2 = J +   32*(I-1)
C  SET UP S(1,1) IN S(1,1,2)
      Q(LOCL+1024) = Q(LOCR2)
  420 CONTINUE
      GO TO 480
  430 CONTINUE
      IF (ICYCLE.GT.2)  GO TO 440
C  S(2,1) IS (N X KREM)   ,  S(2,2) AND S(2,3) ARE (N X N)
      K1 = KREM
      GO TO 450
  440 CONTINUE
C  S(I,I-1) , S(I,I) , S(I,I+1)  ARE (N X N)
      K1 = N
  450 CONTINUE
C  SET UP S(ICYCLE,ICYCLE+1) IN S(1,1,4)
      IF (ICYCLE.EQ.M) GO TO 461
      DO 460 I = 1,N
      DO 460 J = 1,N
      LOCL = J+N*(I-1)
      LOCR1 = J + 32*(I-1+K1+N)
      Q(LOCL+3072) = Q(LOCR1)
  460 CONTINUE
  461 CONTINUE
C  SET UP S(ICYCLE,ICYCLE) IN S(1,1,3)
      DO 468  I=1,N
      DO 468  J=1,N
      LOCR2 = J+32*(I-1+K1)
      LOCL = J + N * (I-1)
      Q(LOCL+2048) = Q(LOCR2)
  468 CONTINUE
C  MOVE  S(ICYCLE,ICYCLE) TO S(1,1,2)
      NSQ = N**2
      DO 470  II=1,NSQ
      Q(II+1024) = Q(II+2048)
  470 CONTINUE
C  REARRANGE S(ICYCLE,ICYCLE-1) WITHIN S(1,1,1)
      KLOC = 0
      DO 480 I=2,K1
      KLOC = KLOC+N
      DO 480 J = 1,N
      Q(KLOC+J) = S(J,I,1)
```

```
  480 CONTINUE
      RETURN
  999 CONTINUE
      PRINT 1000,ICYCLE
 1000 FORMAT (H1,35H  BANDWIDTH EXCEEDED, PARTITION ROW,I3)
      IROW1 = II1 + ISHIFT
      JROW1 = II2 + JSHIFT
      PRINT 1001, IROW1, JROW1
 1001 FORMAT(1H ,4HROW=,I3,6X,7HCOLUMN=,I3)
      PRINT 9120, II1, II2, ISHIFT, JSHIFT
 9120 FORMAT(1H ,4HII1=,I3,4HII2=,I3,7HISHIFT=,I3,7HJSHIFT=,I3)
      PRINT 9121, K1, K2, KREM, N, M
 9121 FORMAT(1H ,3HK1=,I3,3HK2=,I3,5HKREM=,I3,2HN=,I3,2HM=,I3)
      PRINT 9122,I, J, IPL, IMR, IS, IT
 9122 FORMAT(1H ,2HI=,I3,2HJ=,I3,4HIPL=,I3,4HIMR=,I3,3HIS=,I3,3HIT=,I3)
      STOP
      END



      SUBROUTINE MXMULT(A,B,C,M,N,K)
C
C   THIS SUBROUTINE MULTIPLIES MATRIX A BY MATRIX B  AND STORES THE
C   PRODUCT IN C.    (C CANNOT BE THE SAME AS A OR B.)
C
C   A IS (M X N)
C   B IS (N X K)
C   C IS (M X K)
C
      DIMENSION  A(M,N) , B(N,K) , C(M,K)
C
      DO 1  I=1,M
      DO 1  L=1,K
      C(I,L) = 0.
      DO 1  J=1,N
      C(I,L) = C(I,L) + A(I,J) * B(J,L)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MXSUB(A,B,C,M,N)
C
C  THIS SUBROUTINE SUBTRACTS MATRIX B FROM MATRIX A,STORES RESULT IN C
C
C  A, B, AND C ARE (M X N)         (C CAN BE THE SAME AS A OR B)
C
      DIMENSION   A(M,N) , B(M,N) ,C(M,N)
C
      DO 1  I=1,M
      DO 1  J=1,N
      C(I,J) = A(I,J) - B(I,J)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MXCON(A,B,X,M,N)
C
C  THIS SUBROUTINE MULTIPLIES MATRIX A (MXN) BY CONSTANT X, RESULT IN B
C  A MAY BE SAME AS B.
C  THIS SUBROUTINE MULTIPLIES MATRIX A (MXN) BY CONSTANT X, RESULT IN B
C  A MAY BE SAME AS B.
      DIMENSION  A(M,N) , B(M,N)
      DO 1 I=1,M
      DO 1 J=1,N
      B(I,J)= X*A(I,J)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INVERT(B,K,K2,XMIN,FLAG)
C  THIS SUBROUTINE SETS UP A UNIT MATRIX ADJACENT TO B(K,K)
C  ELEMENTARY ROW OPERATIONS ARE THEN PERFORMED ON THE NEW K X 2K MATRIX
C  TO REDUCE B(K,K) TO A UNIT MATRIX.  THIS WILL PLACE THE INVERSE OF
C  THE MATRIX B(K,K) IN THE RIGHT HALF OF B(K,2K)
C  ON EXIT, THE INVERSE OF B REPLACES B
C   B9+S N ARRAY OF 2*K**2 LOCATIONS CONTAINING THE MATRIX
C  K IS THE DIMENSION OF THE SQUARE MATRIX B
C  K2 IS 2*K
C  XMIN IS THE SMALLEST ALLOWABLE MAGNITUDE OF THE PIVOT
C  FLAG WILL BE RETURNED AS 0. IF THE INVERSION WENT OFF OK
C  FLAG WILL BE RETURNED AS 10. IF A PIVOT ELEMENT WAS TOO SMALL
C  FLAG SHOULD BE TESTED AFTER EACH CALL TO THIS ROUTINE
C
      DIMENSION B(K,K2)
C
      FLAG = 0.
C
C  SET UP UNIT MATRIX
C
      IF(K.GT.1)  GO TO 20
      IF(ABS(B(1,1)).LT.XMIN)  GO TO 10
      B(1,1) = 1./B(1,1)
      RETURN
   20 CONTINUE
      DO 1  I=1,K
      DO 1  J=1,K
      B(I,K+J) = 0.
      IF(I.EQ.J)  B(I,K+J) = 1.
    1 CONTINUE
C
C  FIND LEADING ELEMENT WITH GREATEST MAGNITUDE
C
      DO 6  J=1,K
      M = J
      N = J+1
      IF(J.EQ.K) GO TO 21
      DO 2  L=N,K
      IF (ABS(B(M,J)).LT.ABS(B(L,J)))  M=L
    2 CONTINUE
   21 CONTINUE
      IF (ABS(B(M,J)).LT.XMIN)  GO TO 10
      IF(J.EQ.K) GO TO 31
C
C  INTERCHANGE JTH AND MTH ROWS
C
      DO 3  L=J,K2
      D = B(J,L)
      B(J,L) = B(M,L)
      B(M,L) = D
    3 CONTINUE
   31 CONTINUE
C
C  ZERO OUT PIVOTAL JTH COLUMN,  SKIPPING PIVOTAL JTH ELEMENT
C
C  DIVIDE JTH ROW BY PIVOT
C
      DO 4  M=N,K2
      B(J,M) = B(J,M) / B(J,J)
    4 CONTINUE
      DO 6  M=1,K
```

```
C
C   M DETERMINES ROW BEING MODIFIED. ONE WHOLE ROW AT A TIME
C
      IF ( M.EQ.J )    GO TO 6
      DO 5  L=N.K2
C
C   L DETERMINES ELEMENT IN THE MTH ROW
C
      B(M.L) = B(M.L) - B(M.J) * B(J.L)
    5 CONTINUE
    6 CONTINUE
C
C   INVERSE OF B IS NOW IN RIGHT HALF OF B(K.K2)
C   NOW MOVE B INVERSE TO WHERE B WAS
      DO 7  I=1.K
      DO 7  J=1.K
      B(I.J) = B(I.J+K)
    7 CONTINUE
      RETURN
   10 FLAG = 10.
      RETURN
      END
```

FUNDAMENTAL CASES FIVE AND FOUR (LONGITUDINAL LOADING)

FUNDAMENTAL CASE 5

```
      PROGRAM LONTUD
C
C      LONTUD   CASE 5
C
C
C   THIS PROGRAM SOLVES THE LONGTUDINAL SHEAR PROBLEM
      COMMON /1/ X(100), Y(100), NDIR(200,4),M,N,KREM,NN,NT,G(30),
     1            CAY(3,3,201),CENT(200,2),AG(2), CDA(2,3,200),
     2            NTRI(100,10),NUD, NKD, S(30,30,5),FD(30)  , SP(30,15)
      COMMON/2/ VF
      DIMENSION DUM(10) , DMXT(6),  DMX(6)
      DATA  (DMX = 0.,0.,1.,0.,0.,1.) ,(DMXT = 0.,1.,0.,0.,0.,1.)
C    NDIR(I,4) - MATERIAL, 1 OR 2
C    NDIR(I,J) - JTH DIRECTION/NODE IN ITH TRIANGLE ,(J=1,3)
C    NT  -  NUMBER TRIANGLES
C    NN  -  NUMBER NODES/DIRECTIONS
C
C
C
C    NUD -  NUMBER UNKNOWN DISPLACEMENTS
C    NKD -  NUMBER   KNOWN DISPLACEMENTS
      REWIND 20                                                    CASE 5
      CALL TAPESKIP(20,6,0)
      CALL TAPESKIP(20,4,0)
   30 READ 1000,DUM
      PRINT 1001, DUM
      READ 1002,AG(1),AG(2),VF
      IF(VF.EQ.0.) GO TO 950                                       CASE 5
      PRINT 1003,AG(1),AG(2),VF
      M = 5
      N = 17
      KREM = 15
      NT = 158
      NN =  97
      NUD = 83
      NKD = 14
      CALL CONFIG
      PRINT 1009
      DO 100 I=1,14
      J = 83+I
      PRINT 1010,J,FD(I)
  100 CONTINUE
      LINE = 4
      PRINT 1004,(J,J=1,3),(J,J=1,3)
      DO 130 I=1,157,2
      IF(LINE.LT.54) GO TO 120
      LINE = 4
      PRINT 1004,(J,J=1,3),(J,J=1,3)
  120 LINE = LINE + 2
      II=I+1
  130 PRINT 1005,I, (NDIR(I,J),J=1,4),II,(NDIR(II,J),J=1,4)
C
C   PRINT OUT NODE INFORMATION
C
      I = 1
  190 CONTINUE
      LINE = 4
      PRINT 1007
  200 CONTINUE
      IF(I.NE.97) GO TO 210
      PRINT 1008, I, X(I), Y(I)
      GO TO 215
```

```
  210 IP2 = I+2
      PRINT 1018,((J,X(J),Y(J)),J=I,IP2)
  215 I=I+3
      IF(I.GT.NN) GO TO 220
      LINE = LINE + 2
      IF(LINE.GT.56) GO TO 190
      GO TO 200
  220 CONTINUE
C
C   DERIVE ALL MATRICES SMALL K
C
      DO 300 J=1,NT
      CALL KSMALL(J)
  300 CONTINUE
      DO 400 I=1,NN
      NTRI(I,10)= 0
  400 CONTINUE
      DO 500 I = 1,NT
      DO 500 J = 1,3
      K = NDIR(I,J)
      NTRI(K,10) = NTRI(K,10) + 1
      L = NTRI(K,10)
      NTRI(K,L) = I
  500 CONTINUE
C
      CALL PUCHOL
      GO TO 30
 1000 FORMAT(10A8)
 1001 FORMAT(1H1,10A8)
 1002 FORMAT(3E15.4)
 1003 FORMAT(30X,3HG 1,20X,3HG 2,20X,3HV F/10X,3(8X,E15.5))
 1004 FORMAT(1H1, 8HTRIANGLE,3(2X,5HNODE ,I1),4X,8HMATERIAL,10X,
     1              8HTRIANGLE,3(2X,5HNODE ,I1),4X,8HMATERIAL/)
 1005 FORMAT(1X,2(5X,I3,5X,I3,5X,I3,5X,I3 ,9X,I3,10X)/)
 1007 FORMAT(1H1,3(4HNODE,12X,1HX,12X,1HY,8X)/)
 1008 FORMAT(2X,I3,2(3X,F10.5)/)
 1009 FORMAT(/////,4X,15HNON-ZERO KNOWNS,/)
 1010 FORMAT(5H ROW(,I2,2H)=,F11.4)
 1018 FORMAT(1X,3(1X,I3,3X,F10.5,3X,F10.5,8X)/)
  950 CONTINUE                                                    CASE 5
      REWIND 20                                                   CASE 5
      STOP
      END
```

```
      SUBROUTINE CONFIG
C
C
C   CONFIG   CASE 5
C
C   THIS SUBROUTINE GENERATES  THE COORDINATE AND FORCE CONFIGURATION
C   FOR THE PROGRAM LONTUD
C
      COMMON /1/ X(100), Y(100), NDIR(200,4),M,N,KREM,NN,NT,G(30),
     1           CAY(3,3,201),CENT(200,2),AG(2), CDA(2,3,200),
     2          NTRI(100,10),NUD, NKD, S(30,30,5),FD(30)  , SP(30,15)
      COMMON/2/ VF
      DIMENSION JNDIR(158),KNDIR(158),LNDIR(158)
      DATA  (RAD = 57.29578)
      DATA (JNDIR =
     1    1, 2, 3, 1, 1, 2, 2, 2, 3, 3, 3, 9, 4, 4, 5, 6, 6, 6, 7, 8,
     2    8, 8, 9,15,10,10,11,12,12,12,13,14,14,15,15,21,16,20,20,20,
     3   21,21,21,25,23,24,24,25,25,28,28,17,17,17,18,18,19,19,22,22,
     4   22,23,23,29,29,30,30,31,31,32,32,33,33,34,34,35,35,36,36,38,
     5   38,39,39,40,40,41,41,42,42,43,43,44,44,45,45,47,47,48,48,49,
     6   50,50,51,51,52,52,53,54,54,59,57,57,57,58,59,59,59,60,61,61,
     7   62,56,63,63,63,63,64,65,65,65,66,67,67,67,69,69,69,69,70,71,
     8   71,71,72,73,73,73,75,75,75,76,77,77,78,79,79,81,81,82)
      DATA (KNDIR =
     1    2, 3,85, 4, 5, 5, 6, 7, 7, 8, 9,86,10,11,11,11,12,13,13,13,
     2   14,15,15,87,16,17,17,17,18,19,19,19,20,14,21,88,28,19,22,23,
     3   23,24,25,89,26,26,27,27,90,29,30,30,31,32,32,33,33,34,34,35,
     4   36,36,37,38,39,39,40,40,41,41,42,42,43,43,44,44,45,45,46,47,
     5   48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,58,61,61,62,62,
     6   62,65,65,66,66,67,67,67,56,97,97,59,60,60,96,95,63,63,63,64,
     7   64,67,95,94,69,70,70,70,71,72,72,72,73,74,94,93,75,76,76,76,
     8   77,78,78,78,79,80,93,92,81,81,81,82,82,82,83,92,91,91)
      DATA (LNDIR =
     1   84,84,84, 5, 2, 6, 7, 3, 8, 9,85,85,11, 5, 6,12,13, 7, 8,14,
     2   15, 9,86,86,17,11,12,18,19,13,14,20,21,21,87,87,17,22,23,21,
     3   24,25,88,88,24,27,25,90,89,30,17,31,32,18,33,19,34,22,35,36,
     4   23,37,26,39,30,40,31,41,32,42,33,43,34,44,35,45,36,46,37,48,
     5   39,49,40,50,41,51,42,52,43,53,44,54,45,55,46,61,48,62,49,50,
     6   65,51,66,52,67,53,54,56,55,96,59,60,58,61,95,63,60,61,64,62,
     7   65,68,94,69,70,64,65,71,72,66,67,73,74,68,93,75,76,70,71,77,
     8   78,72,73,79,80,74,92,81,76,77,82,78,79,83,80,91,82,83)
      DO 10 I=1,36
      NDIR(I,4) = 1
   10 NDIR(159-I,4) = 1
      DO 20 I = 1,86
      NDIR(36+I,4) = 2
   20 CONTINUE
      DO 50  I = 1,7
      FD(I) = 1
   50 FD(I+7) = -1
      DO 100  I = 1, 158
      NDIR(I,1) = JNDIR(I)
      NDIR(I,2) = KNDIR(I)
      NDIR(I,3) = LNDIR(I)
  100 CONTINUE
      S3 = SQRT(3.)
      S302 = S3/2.
      PI = 3.1415927
      R = SQRT(2.*S3*VF/PI)
      X(84) = S302
      Y(84) =  .5
      DO  300  I=1,3
```

```
      X(I) = S302-R/4.*COS(PI*(I-1)/6.)
  300 Y(I)=  .5 -R/4.*SIN(PI*(I-1)/6.)
C
      DO 310  I=1,6
      X(I+3) =   S302 - R/2.* COS(PI*(I-1)/12.)
      Y(I+3) =    .5  - R/2.* SIN(PI*(I-1)/12.)
C
      X(I+9) =   S302 - 3.*R/4.*COS(PI*(I-1)/12.)
      Y(I+9) =    .5  - 3.*R/4.*SIN(PI*(I-1)/12.)
C
      X(I+15) = S302 - R * COS(PI*(I-1)/12.)
  310 Y(I+15) =  .5  - R * SIN(PI*(I-1)/12.)
      DO 320 I =1,7
      X(83+I) = S302
  320 CONTINUE
      DO 330 I =1,5
      Y(83+I)=  .5- (I-1)/4.*R
  330 CONTINUE
      Y(90) = -.5
      Y(89) = (Y(88)+Y(90))/2.
      X(38) = -.5 *TAN(PI/6.)
      Y(38) =  .5
      DX = (1.-R)/(2.*COS(PI/6.))
      X(29) = X(38) + DX
      Y(29) = .5
      X(28) = (X(29)+X(16))/2.
      Y(28) = .5
      DO 340  I =1,4
      X(I+38) = (4-I) * X(38)/4.
  340 Y(I+38) = (4-I) * Y(38)/4.
      DELX = X(39) - X(38)
      DELY = Y(39) - Y(38)
      DO  350  I = 1,8
      X(I+29) = X(I+28) + DELX
  350 Y(I+29) = Y(I+28) + DELY
      X(26) = X(37)+(X(90)-X(37))/3.
      Y(26) = -.5
      X(27) = 2.*X(26) - X(37)
      Y(27) = -.5
      X(22) = X(19) + (X(26)-X(19)) /3.
      Y(22) = Y(19) + (Y(26)-Y(19)) /3.
      X(23) = 2.*X(22)-X(19)
      Y(23) = 2.*Y(22)-Y(19)
      X(24) = X(23)+(X(89)-X(23))/3.
      Y(24) = Y(23)+(Y(89)-Y(23))/3.
      X(25) = 2.*X(24)-X(23)
      Y(25) = 2.*Y(24)-Y(23)
C
      DO 360 I= 43,83
      X(I) = -X(84-I)
  360 Y(I) = -Y(84-I)
C
      DO 370 I=  1,7
      X(90+I) = -X(83+I)
  370 Y(90+I) = -Y(83+I)
C
      RETURN
      END
```

```
      SUBROUTINE FINAL
C
C     FINAL   CASE 5
C
C
C  THIS SUBROUTINE CALCULATES STRESSES FOR PROGRAM LONTUD
C
C
      COMMON /1/ X(100), Y(100), NDIR(200,4),M,N,KREM,NN,NT,G(30),
     1           CAY(3,3,201),CENT(200,2),AG(2), CDA(2,3,200),
     2          NTRI(100,10),NUD, NKD, S(30,30,5),FD(30)   , SP(30,15)
      COMMON/2/ VF
      DIMENSION SIG(4,158)
      EQUIVALENCE  (SIG,STR)
      DIMENSION  SIGOUT(632)
      DIMENSION  DEL(200), DX(3),  STR(4,200)
      DATA  (PI=3.1415927)
      R=SQRT(3.)
      R=SQRT(2.*R*VF/PI)
      DEG = 57.29578
      DO 10 J = 1,KREM
      DEL(J) = SP(J,1)
   10 CONTINUE
      KLOC = KREM - N
      DO 20 I = 2,M
      KLOC = KLOC +N
      DO 20 J =1,N
      DEL(KLOC+J) = SP(J,I)
   20 CONTINUE
      DO 25 I = 1, NKD
   25 DEL(NUD+I) = FD(I)
      PRINT 1010
      NDEL = NN / 7
      JCNT = 0
      DO 35 J = 1,NDEL
      JCNT = JCNT + 1
      IF(JCNT.LE.18) GO TO 30
      PRINT 1010
      JCNT = 0
   30 JFIR = 7*(J-1)+1
      JLAST = JFIR + 6
      PRINT 1011, (K,K=JFIR,JLAST)
      PRINT 1012, (DEL(K),K=JFIR,JLAST)
   35 CONTINUE
      LOC1 = 7*NDEL+1
      IF(LOC1.GT.NN) GO TO 40
      PRINT 1011, (K,K=LOC1,NN)
      PRINT 1012, (DEL(K),K=LOC1,NN)
   40 CONTINUE
C
C  FIND STRESSES
C
      DO 100 I=1,NT
      I1 = NDIR(I,1)
      I2 = NDIR(I,2)
      I3 = NDIR(I,3)
      DX(1) = DEL(I1)
      DX(2) = DEL(I2)
      DX(3) = DEL(I3)
      CALL GMMULT(CDA(1,1,I),DX,STR(1,I),2,3,1)
      STR(3,I) = SQRT( STR(1,I)**2 + STR(2,I)**2)
  100 STR(4,I) = DEG * ATAN(STR(2,I) / STR(1,I) )
```

```
      TAUXZ = R/4.*(SIG(1,3)+SIG(1,12)+SIG(1,24)+SIG(1,36))
     1      + (1.-R)/2.*(SIG(1,44)+SIG(1,49))
      TAUXZ = 1./TAUXZ
      DO 110 J=1,158
      STR(3,J) = STR(3,J)*TAUXZ
      DO 110 I=1,2
      K= (J-1)*2+I
      SIGOUT(K) = STR(I,J)*TAUXZ
  110 STR(I,J) = SIGOUT(K)
C
      LINE = 4
      PRINT 1005
      DO 150 I =1,NT
      IF(LINE.LT.54) GO TO 140
      LINE = 4
      PRINT 1005
  140 LINE = LINE + 2
  150 PRINT 1006,I,(CENT(I,J),J=1,2),(STR(J,I),J=1,4)
      PRINT 8787,TAUXZ
 8787 FORMAT(//23H NORMALIZATION FACTOR =,2X,E15.7)
      WRITE (20) (SIGOUT(I),I=1,158)                                    CASE 5
 1005 FORMAT(9H1TRIANGLE,4X10HCENTROID X,4X,10HCENTROID Y,6X,8HSIGMA XZ,
     1       6X,8HSIGMA YZ,6X,8HGRADIENT,9X,5HALPHA/)
 1006 FORMAT(6X,I3,2(4X,F10.4),4(2X,E12.5)/)
 1010 FORMAT(1H1,50X,13HDISPLACEMENTS,//)
 1011 FORMAT(1H ,7(8X,4HDEL(,I3,1H) ) )
 1012 FORMAT(1H ,7(2X,E14.7)/)
      RETURN
      END
```

FUNDAMENTAL CASE 4

```
      PROGRAM LONTUD
C
C    LONTUD  CASE 4
C
C
C   THIS PROGRAM SOLVES THE LONGTUDINAL SHEAR PROBLEM
      COMMON /1/ X(100), Y(100), NDIR(200,4),M,N,KREM,NN,NT,G(30),
     1           CAY(3,3,201),CENT(200,2),AG(2), CDA(2,3,200),
     2           KTRI(100,10),NUD, NKD, S(30,30,5),FD(30)  , SP(30,15)
      COMMON/2/ VF
      DIMENSION DUM(10) , DMXT(6),  DMX(6)
      DATA  (DMX = 0.,0.,1.,0.,0.,1.) ,(DMXT = 0.,1.,0.,0.,0.,1.)
C    NDIR(I,4) - MATERIAL, 1 OR 2
C    NDIR(I,J) - JTH DIRECTION/NODE IN ITH TRIANGLE ,(J=1,3)
C    NT  -  NUMBER TRIANGLES
C    NN  -  NUMBER NODES/DIRECTIONS
C
C
C
C    NUD -  NUMBER UNKNOWN DISPLACEMENTS
C    NKD -  NUMBER   KNOWN DISPLACEMENTS
      REWIND 20                                                  CASE 4
      CALL TAPESKIP(20,6,0)
      CALL TAPESKIP(20,5,0)
   10 READ 1000,DUM
      PRINT 1001, DUM
      READ 1002,AG(1),AG(2),VF
      IF(VF.EQ.0.) GO TO 950                                     CASE 4
      PRINT 1003,AG(1),AG(2),VF
      M = 5
      N=15
      KREM = 13
      NUD = 73
      NT = 158
      NN =  97
      NKD = 24
      CALL CONFIG
      PRINT 1009
      DO 100 I=1,24
      J = 73+I
      PRINT 1010,J,FD(I)
  100 CONTINUE
      LINE = 4
      PRINT 1004,(J,J=1,3),(J,J=1,3)
      DO 130 I=1,157,2
      IF LINE.LT.54) GO TO 120
      LINE = 4
      PRINT 1004,(J,J=1,3),(J,J=1,3)
  120 LINE = LINE + 2
      II=I+1
  130 PRINT 1005,I, (NDIR(I,J),J=1,4),II,(NDIR(II,J),J=1,4)
C
C    PRINT OUT NODE INFORMATION
C
      I = 1
  190 CONTINUE
      LINE = 4
      PRINT 1007
  200 CONTINUE
      IF(I.NE.97) GO TO 210
      PRINT 1008, I, X(I), Y(I)
      GO TO 215
```

```
  210 IP2 = I+2
      PRINT 1018,((J,X(J),Y(J)),J=I,IP2)
  215 I=I+3
      IF(I.GT.NN) GO TO 220
      LINE = LINE + 2
      IF(LINE.GT.56) GO TO 190
      GO TO 200
  220 CONTINUE
C
C   DERIVE ALL MATRICES SMALL K
C
      DO 300 J=1,NT
      CALL KSMALL(J)
  300 CONTINUE
      DO 400 I=1,NN
      NTRI(I,10)= 0
  400 CONTINUE
      DO 500 I = 1,NT
      DO 500 J = 1,3
      K = NDIR(I,J)
      NTRI(K,10) = NTRI(K,10) + 1
      L = NTRI(K,10)
      NTRI(K,L) = I
  500 CONTINUE
C
      CALL PUCHOL
      GO TO 10
  950 CONTINUE                                                          CASE 6
      REWIND 20                                                         CASE 6
      STOP                                                              CASE 6
 1000 FORMAT(10A8)
 1001 FORMAT(1H1,10A8)
 1002 FORMAT(3E15.4)
 1003 FORMAT(30X,3HG 1,20X,3HG 2,20X,3HV F/10X,3(8X,E15.5))
 1004 FORMAT(1H1, 8HTRIANGLE,3(2X,5HNODE ,I1),4X,8HMATERIAL,10X,
     1             8HTRIANGLE,3(2X,5HNODE ,I1),4X,8HMATERIAL/)
 1005 FORMAT(1X,2(5X,I3,5X,I3,5X,I3,5X,I3 ,9X,I3,10X)/)
 1007 FORMAT(1H1,3(4HNODE,12X,1HX,12X,1HY,8X)/)
 1008 FORMAT(2X,I3,2(3X,F10.5)/)
 1009 FORMAT(/////,4X,15HNON-ZERO KNOWNS,/)
 1010 FORMAT(5H ROW(,I2,2H)=,F11.4)
 1018 FORMAT(1X,3(1X,I3,3X,F10.5,3X,F10.5,8X)/)
      END
                                                                    890 CARDS
```

```
      SUBROUTINE CONFIG
C
C    CONFIG    CASE 6
C
C
C    THIS SUBROUTINE GENERATES  THE COORDINATE AND FORCE CONFIGURATION
C    FOR THE PROGRAM LONTUD
C
      COMMON /1/ X(100), Y(100), NDIR(200,4),M,N,KREM,NN,NT,G(30),
     1           CAY(3,3,201),CENT(200,2),AG(2), CDA(2,3,200),
     2          NTRI(100,10),NUD, NKD, S(30,30,5),FD(30)   , SP(30,15)
      COMMON/2/ VF
      DIMENSION JNDIR(158),KNDIR(158),LNDIR(158)
      DATA  (RAD = 57.29578)
      DATA (JNDIR =
     1   1, 1, 2, 4, 1, 1, 1, 1, 2, 2, 2, 3,10, 4, 4, 5, 5, 5, 6, 7,
     2   7, 7, 8, 9,16,10,10,11,11,11,12,13,13,13,14,15,16,18,19,19,
     3  20,20,20,21,23,24,24,25,25,27,16,16,16,16,17,17,18,22,22,22,
     4  22,23,23,34,27,27,27,28,28,29,29,30,30,31,31,32,32,33,33,41,
     5  34,34,34,35,35,36,36,37,37,38,38,39,39,40,40,51,41,41,41,42,
     6  43,43,44,44,45,45,46,47,47,48,49,49,50,50,48,49,49,50,51,51,
     7  52,58,53,54,54,54,55,56,56,56,57,58,58,58,59,60,60,60,61,62,
     8  62,62,63,64,64,64,65,66,66,67,68,68,69,70,70,71,72,73 )
      DATA (KNDIR =
     1  74, 2, 3,75,75, 4, 5, 6, 6, 7, 8, 8,76,76,10,10,11,12,12,12,
     2  13,14,14,14,77,77,16,16,17,18,18,18,19,20,20,20,78,22,22,23,
     3  23,24,25,25,88,88,87,87,86,79,79,27,28,29,29,30,30,18,31,32,
     4  33,33,89,80,80,34,35,35,36,36,37,37,38,38,39,39,40,40,90,81,
     5  81,41,42,42,43,43,44,44,45,45,46,46,47,47,91,82,82,51,52,52,
     6  52,56,56,57,57,58,58,58,92,49,84,50,83,51,53,53,54,54,54,55,
     7  55,93,59,59,60,61,61,61,62,63,63,63,64,94,65,65,66,67,67,67,
     8  68,69,69,69,70,95,71,71,72,72,72,73,73,73,96,97,97,97 )
      DATA (LNDIR =
     1  75,74,74,76, 4, 5, 6, 2, 7, 8, 3, 9,77,10, 5,11,12, 6, 7,13,
     2  14, 8, 9,15,78,16,11,17,18,12,13,19,20,14,15,21,79,19,23,20,
     3  24,25,21,26,24,87,25,86,26,80,27,28,29,17,30,18,31,31,32,33,
     4  23,89,88,81,34,35,28,36,29,37,30,38,31,39,32,40,33,90,89,82,
     5  41,42,35,43,36,44,37,45,38,46,39,47,40,91,90,83,51,52,42,43,
     6  56,44,57,45,58,46,47,92,91,85,85,84,84,83,49,54,50,51,55,52,
     7  56,92,54,60,61,55,56,62,63,57,58,64,94,93,60,66,67,61,62,68,
     8  69,63,64,70,95,94,66,72,67,68,73,69,70,96,95,72,73,96 )
      DO 10 I=1,36
      NDIR(I,4) = 1
   10 NDIR(159-I,4) = 1
      DO 20 I = 1,86
      NDIR(36+I,4) = 2
   20 CONTINUE
      DO 50 I = 1, 12
      FD(I) = 1.
   50 FD(I+12) = -1.
      DO 100  I = 1, 158
      NDIR(I,1) = JNDIR(I)
      NDIR(I,2) = KNDIR(I)
      NDIR(I,3) = LNDIR(I)
  100 CONTINUE
      S3 = SQRT(3.)
      S302 = S3/2.
      PI = 3.1415927
      R = SQRT(2.*S3*VF/PI)
      DO  300  I=1,3
      X(I) = S302-R/4.*COS(PI*I/6.)
  300 Y(I) = .5 - R/4.*SIN(PI*I/6.)
```

```
C
      DO 310  I=1,6
      X(I+3) = S302 - R/2.*COS(PI*I/12.)
      Y(I+3) =  .5  - R/2.*SIN(PI*I/12.)
C
      X(I+9) = S302 - 3.*R/4.*COS(PI*I/12.)
      Y(I+9) =  .5  - 3.*R/4.*SIN(PI*I/12.)
C
      X(I+15)= S302 - R*COS(PI*I/12.)
  310 Y(I+15)= .5 - R* SIN(PI*I/12.)
      DO 320 I = 1,5
      X(I+73) = S302 - R*(I-1)/4.
  320 Y(I+73) = .5
      DO 330 I=1,4
  330 Y(85+I) = -.5
      X(86) = S302
      X(26) = S302
      Y(26) =(Y(21)+ Y(86)) /2.
      Y(81) = .5
      Y(80) = .5
      Y(79) = .5
      X(81) = -.5 * TAN(PI/6.)
      DX = (1.-R) / (2.*COS(PI/6.))
      X(80) = X(81) + DX
      X(79) = (X(80) + X(78))/2.
      DO 340 I=1,4
      X(I+33) = (4-I)* X(81)/4.
  340 Y(I+33) = (4-I)* Y(81)/4.
      DELX = X(34)-X(81)
      DELY = Y(34)-Y(81)
      DO 350 I = 1,7
      X(I+26) = X(80)+DELX*I
  350 Y(I+26) = Y(80)+DELY*I
      X(89) = X(33)+DELX
      X(88) = X(89)+ (X(86)-X(89))/3.
      X(87) = 2.*X(88)-X(89)
      X(22) = X(18) + (X(88)-X(18)) /3.
      Y(22) = Y(18) + (Y(88)-Y(18)) /3.
      X(23) = 2.*X(22) - X(18)
      Y(23) = 2.*Y(22) - Y(18)
      X(24) = X(23)+(X(26)-X(23))/3.
      Y(24) = Y(23)+(Y(26)-Y(23))/3.
      X(25) = 2.*X(24) -X(23)
      Y(25) = 2.*Y(24) -Y(23)
      DO 360 I= 38,73
      X(I) = -X(74-I)
  360 Y(I) = -Y(74-I)
      DO 370 I= 1, 4
      X(89+I) = -X(82-I)
      Y(89+I) = -.5
      X(93+I) = -X(78-I)
      Y(93+I) = -.5
      X(81+I) = - X(90-I)
  370 Y(81+I) = +.5
C
      RETURN
      END
```

```
      SUBROUTINE FINAL
C
C     FINAL   CASE 6
C
C
C  THIS SUBROUTINE CALCULATES STRESSES FOR PROGRAM LONTUD
C
C
      COMMON /1/ X(100), Y(100), NDIR(200,4),M,N,KREM,NN,NT,G(30),
     1            CAY(3,3,201),CENT(200,2),AG(2), CDA(2,3,200),
     2           NTRI(100,10),NUD, NKD, S(30,30,5),FD(30)   , SP(30,15)
      COMMON/2/ VF
      DIMENSION   SIG(4,158)
      EQUIVALENCE    (SIG,STR)
      DIMENSION  SIGOUT(632)
      DIMENSION  DEL(200), DX(3),  STR(4,200)
      DATA  (PI=3.1415927)
      R=SQRT(3.)
      R=SQRT(2.*R*VF/PI)
      DEG = 57.29578
      DO 10 J = 1,KREM
      DEL(J) = SP(J,1)
   10 CONTINUE
      KLOC = KREM - N
      DO 20 I = 2,M
      KLOC = KLOC +N
      DO 20 J =1,N
      DEL(KLOC+J) = SP(J,I)
   20 CONTINUE
      DO 25 I = 1, NKD
   25 DEL(NUD+I) = FD(I)
      PRINT 1010
      NDEL = NN / 7
      JCNT = 0
      DO 35 J = 1,NDEL
      JCNT = JCNT + 1
      IF(JCNT.LE.18) GO TO 30
      PRINT 1010
      JCNT = 0
   30 JFIR = 7*(J-1)+1
      JLAST = JFIR + 6
      PRINT 1011, (K,K=JFIR,JLAST)
      PRINT 1012, (DEL(K),K=JFIR,JLAST)
   35 CONTINUE
      LOC1 = 7*NDEL+1
      IF(LOC1.GT.NN) GO TO 40
      PRINT 1011, (K,K=LOC1,NN)
      PRINT 1012, (DEL(K),K=LOC1,NN)
   40 CONTINUE
C
C  FIND STRESSES
C
      DO 100 I=1,NT
      I1 = NDIR(I,1)
      I2 = NDIR(I,2)
      I3 = NDIR(I,3)
      DX(1) = DEL(I1)
      DX(2) = DEL(I2)
      DX(3) = DEL(I3)
      CALL MXMULT(CDA(1,1,I),DX,STR(1,I),2,3,1)
  100 STR(3,I) = SQRT( STR(1,I)**2 + STR(2,I)**2)
      TAUYZ = R/4.*(SIG(2,1)+SIG(2,4)+SIG(2,13)+SIG(2,25))
```

```
     1      + (X(78)-X(79))*(SIG(2,37)+S.G(2,50))
     2      + (X(80)-X(81))*(SIG(2,64)+SIG(2,80))
     3      + (X(82)-X(85))/3.*(SIG(2,96)+SIG(2,111)+SIG(2,113))
      TAUYZ=SQRT(3.)/TAUYZ
      DO 110 J=1,158
      IF(1.E10*ABS(STR(1,J)).LT.ABS(STR(2,J))) GO TO 80
      STR(4,J)=DEG*ATAN(STR(2,J)/STR(1,J))
      GO TO 90
   80 STR(4,J) = 90.
   90 CONTINUE
      STR(3,J) = STR(3,J)*TAUYZ
      DO 110 I=1,2
      K= (J-1)*2+I
      SIGOUT(K) = STR(I,J)*TAUYZ
  110 STR(I,J) = SIGOUT(K)
C
      LINE = 4
      PRINT 1005
      DO 150 I =1,NT
      IF(LINE.LT.54) GO TO 140
      LINE = 4
      PRINT 1005
  140 LINE = LINE + 2
  150 PRINT 1006,I,(CENT(I,J),J=1,2),(STR(J,I),J=1,4)
      PRINT 8787,TAUYZ
 8787 FORMAT(//23H NORMALIZATION FACTOR =,2X,E15.7)
      WRITE (20) (SIGOUT(I),I=1,158)                                  CASE 6
 1005 FORMAT(9H1TRIANGLE,4X10HCENTROID X,4X,10HCENTROID Y,6X,8HSIGMA XZ,
     1      6X,8HSIGMA YZ,6X,8HGRADIENT,9X,5HALPHA/)
 1006 FORMAT(6X,I3,2(4X,F10.4),4(2X,E12.5)/)
 1010 FORMAT(1H1,50X,13HDISPLACEMENTS,//)
 1011 FORMAT(1H ,7(8X,4HDEL(,I3,1H) ) )
 1012 FORMAT(1H ,7(2X,E14.7)/)
      RETURN
      END
```

AUXILIARIES

```
      SUBROUTINE PUCHOL
C
C  THIS ROUTINE SOLVES  SU=G , WHERE S IS A TRI-DIAGONAL MATRIX IN
C  SUBMATRICES, WITH ELEMENTS OF ORDER N.
C  S IS KNOWN
C  SP IS A VECTOR OF DIMENSION (NXM) WHERE M IS THE NUMBER OF DIVISIONS OF S
C  C IS WRITTEN ONTO TAPE AFTER DERIVATION ON THE FORWARD PASS.
C  AND READ BACK IN ON THE BACKSWEEP
C  S(1,1,1)  INITIALLY CONTAINS  S(I,I-1)
C  S(1,1,2)  INITIALLY CONTAINS  S(I,I  )
C  S(1,1,4)  INITIALLY CONTAINS  S(I,I+1)
C  SP CORRESPONDS TO P IN THE WRITEUP BY GATEWOOD ON THE FORWARD PASS.
C  ON THE BACKSWEEP, IT CORRESPONDS TO U.
C
      COMMON /1/ X(100), Y(100), NDIR(200,4),M,N,KREM,NN,NT,G(30),
     1           CAY(3,3,201),CENT(200,2),AG(2), CDA(2,3,200),
     2          NTRI(100,10),NUD, NKD, S(30,30,5),FD(30)  , SP(30,15)
      DIMENSION C(900)
      EQUIVALENCE  (S(3601),C)
      DATA (XLIMIT =1.E-8)
      REWIND 96
      N2 = 2*N
      KREM2 = 2*KREM
      DO 30 ICYCLE =1,M
      IF(ICYCLE-2) 1,2,3
    1 K1 = KREM
      K2 = KREM
      K3 = KREM2
      GO TO 4
    2 K1 = N
      K2 = KREM
      K3 = N2
      GO TO 4
    3 K1 =N
      K2 =N
      K3 =N2
    4 CONTINUE
      K4 = N
      IF(ICYCLE.EQ.M)  K4 = NKD
      CALL KLARGE(ICYCLE,S,S(1,1,2),S(1,1,4),S(1,1,3),K1,K2,K4,NKD)
      IF(ICYCLE.EQ.1) GO TO 10
    5 IF (UNIT,96)  6,7 ,600, 600
    6 GO TO 5
    7 CONTINUE
      CALL MXMULT(S(1,1,1) , S(1,1,5) , S(1,1,3) ,K1,K2,K1)
C
C  S(1,1,5) CONTAINS C FROM LAST CYCLE
C
      CALL MXSUB(S(1,1,2) , S(1,1,3) , S(1,1,2) ,K1,K1)
   10 CONTINUE
C
C  B(I,I)  NOW IN S(1,1,2)
C
      CALL INVERT(S(1,1,2) , K1 , K3, XLIMIT , FLAG)
      IF (FLAG.NE.0.) GO TO 500
C
C  INVERSE OF B(I,I) NOW IN  S(1,1,2)
C
      IF (ICYCLE.EQ.1) GO TO 20
      CALL MXMULT(S(1,1,1) ,SP(1,ICYCLE-1) , S(1,1,3) , N, K2 , 1)
      CALL MXSUB (G  ,  S(1,1,3), G, N, 1)
   20 CONTINUE
```

```
      CALL MXMULT [illegible] , S ,SP(1, ICYCLE  ) , K1 , K1 ,  1)
      IF (ICYCLE.GE.M) GO TO 35
      CALL MXMULT (S(1,1,2) , S(1,1,4) , S(1,1,5), K1 , K1 , N)
      NSQ = K1*N
      BUFFER OUT  (96,1)  (C(1), C(NSQ ))
   30 CONTINUE
   35 CONTINUE
C
C  NOW IN BACKSWEEP, SOLVING FOR U
C
      DO 60 I = 2,M
      JCYCLE = M-I+1
      IF(JCYCLE.GT.1)  GO TO 36
      K1 = KREM
      GO TO 37
   36 K1 = N
   37 CONTINUE
      NSQ = K1*N
      IF (JCYCLE.EQ.M-1)  GO TO 41
      BACKSPACE 96
   41 CONTINUE
      BACKSPACE 96
      BUFFER IN   (96,1)  (C(1),C(NSQ ))
   42 IF(UNIT,96) 43, 44,700,700
   43 GO TO 42
   44 CONTINUE
C
C  U(M)=SP(M) , CONSIDER FIRST (M-1)TH CYCLE
C
      CALL MXMULT(S(1,1,5) ,SP(1,JCYCLE+1),S(1,1,1),K1,N, 1)
      CALL MXSUB(SP(1,JCYCLE), S(1,1,1) ,SP(1,JCYCLE), K1 ,  1)
   60 CONTINUE
C
C  U(NS,I) NOW STORED IN SP(N,I) , I=1,M
C
      CALL FINAL
      RETURN
  500 CONTINUE
      PRINT 1000 , ICYCLE
      STOP
  600 CONTINUE
      PRINT 1001,ICYCLE
      STOP
  700 CONTINUE
      PRINT 1002, JCYCLE
 1000 FORMAT (31H1COULD NOT INVERT MATRIX IN ROW,I2)
 1001 FORMAT(37H1        ERROR READING C INTO CORE ON I2, 7HTH ROW.)
 1002 FORMAT(37H1        ERROR WRITING C ONTO TAPE ON I2, 7HTH ROW.)
      STOP
      END
```

```
      SUBROUTINE KSMALL(I)
C
C  THIS SUBROUTINE DERIVES THE MATRIX  SMALL K  AND THE MATRIX
C  C*D* A**(-1)        FOR THE TRIANGLE I.
C
      COMMON /1/ X(100), Y(100), NDIR(200,4),M,N,KREM,NN,NT,G(30),
     1            CAY(3,3,201),CENT(200,2),AG(2), CDA(2,3,200),
     2           NTRI(100,10),NUD, NKD, S(30,30,5),FD(30)   , SP(30,15)
      DIMENSION  D(6), AINV(3,3), C(4) ,DMXT(6)
      DATA   (DMXT = 0.,1.,0.,0.,0.,1.)
      DATA   (D=0.,0.,1.,0.,0.,1.) , (C = 0.,0.,0.,0.)
      AINV(1,1) =1.
      AINV(1,2) =0.
      AINV(1,3) =0.
C
      IN1 = NDIR(I,1)
      IN2 = NDIR(I,2)
      IN3 = NDIR(I,3)
      XI2 = X(IN2) - X(IN1)
      XI3 = X(IN3) - X(IN1)
      ETA2= Y(IN2) - Y(IN1)
      ETA3= Y(IN3) - Y(IN1)
C
      CENT(I,1) = (X(IN1) +X(IN2) + X(IN3) ) / 3.
      CENT(I,2) = (Y(IN1) +Y(IN2) +Y(IN3) ) / 3.
C
      DELTA = XI2*ETA3 - XI3*ETA2
      IF(ABS(DELTA).GT.1.E-10)  GO TO 77777
      PRINT 88888,I
88888 FORMAT(1H1,3HI =,I3)
77777 CONTINUE
C
      AINV(2,2) = ETA3/DELTA
      AINV(2,3) = -ETA2/DELTA
      AINV(2,1) = -(AINV(2,2) + AINV(2,3) )
      AINV(3,2) = -XI3/DELTA
      AINV(3,3) =  XI2/DELTA
      AINV(3,1) = -(AINV(3,2) + AINV(3,3) )
C
      IMAT = NDIR(I,4)
      C(1) =AG(IMAT)
      C(4) = C(1)
      CALL MXMULT(D,AINV,CAY(1,1,I),2,3,3)
      CALL MXMULT(C,CAY(1,1,I),CDA(1,1,I),2,2,3)
      DO 100 J=2,3
      JJ=J-1
      DO 100 K=1,JJ
      TEMP = AINV(J,K)
      AINV(J,K) = AINV(K,J)
  100 AINV(K,J) = TEMP
      OMEGA = DELTA / 2.
      CALL MXMULT(DMXT,CDA(1,1,I),CAY(1,1,I+1),3,2,3)
      CALL MXMULT(AINV,CAY(1,1,I+1),CAY(1,1,I),3,3,3)
      DO 110 J = 1,3
      DO 110 K = 1,3
      CAY(J,K,I) = CAY(J,K,I)* OMEGA
  110 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE KLARGE(I,SIM1,SI,SIP1,CAY12, I1,I2,I3,I4)
C
C  THIS SUBROUTINE IS CALLED BY SUBROUTINE PUCHOL TO GENERATE THE I-TH
C  ROW OF SUBMATRICES S(I,I-1), S(I,I), S(I,I+1) AND THE PART OF THE
C  INDEPENDENT VECTOR REQUIRED BY THE CHOLESKI PROCESS ON THE I-TH PASS
C
      COMMON /1/ X(100), Y(100), NDIR(200,4),M,N,KREM,NN,NT,G(30),
     1           CAY(3,3,201),CENT(200,2),AG(2), CDA(2,3,200),
     2          NTRI(100,10),NUD, NKD, S(30,30,5),FD(30)  , SP(30,15)
      DIMENSION SIM1(I1,I2), SI(I1,I1), SIP1(I1,I3) , CAY12(I1,I4)
      KS2= 4
      IF(I.GT.1) GO TO 10
      KS1= 2
      IS = 0
      JS = - I2
      GO TO 40
   10 IF(I.GT.2) GO TO 20
      JS = 0
      GO TO 30
   20 JS = KREM + (I-3)*N
   30 IS = KREM + (I-2)*N
C
C  IS - ROW BIAS
C  JS - COLUMN BIAS
C
      KS1 = 1
   40 JS1 = JS
      JS2 = JS1 + I2
      JS3 = JS2 + I1
      JS4 = JS3 + I3
      IROW1 = 1 + IS
      IROW2 = I1 + IROW1 - 1
      DO 300 KSS =KS1,KS2
C
      GO TO (100,110,120,130),KSS
  100 ICOL1 = JS1 +1
      ICOL2 = JS2
      GO TO  150
  110 ICOL1 = JS2+1
      ICOL2 = JS3
      GO TO  150
  120 IF(I.EQ.M) GO TO 300
      ICOL1 = JS3+1
      ICOL2 = JS4
      GO TO 150
  130 ICOL1 =  NUD + 1
      ICOL2 = NN
  150 CONTINUE
      DO 300 II =IROW1,IROW2
      DO 300 JJ =ICOL1,ICOL2
      TEMP = 0.
      NTRI9 = NTRI(II,10)
      NTRJ9 = NTRI(JJ,10)
C
C  NTRI9 - NUMBER TRIANGLES TOUCHING NODE II
C  NTRJ9 - NUMBER TRIANGLES TOUCHING NODE JJ
C
      DO 200 K = 1,NTRI9
      NTRII = NTRI(II,K)
      DO 200 KK = 1,NTRJ9
      IF(NTRII.NE.NTRI(JJ,KK))GO TO 200
      DO 180 L = 1,3
```

```
      IF(II.EQ.NDIR(NTRII.L))IK = L
      IF(JJ.EQ.NDIR(NTRII.L))JK = L
  180 CONTINUE
      TEMP = TEMP + CAY(IK.JK.NTRII)
  200 CONTINUE
      GO TO(210.220.230.240). KSS
  210 SIM1(II-IS.JJ-JS1) = TEMP
      GO TO 300
  220 SI(II-IS.JJ-JS2) = TEMP
      GO TO 300
  230 SIP1(II-IS.JJ-JS3) = TEMP
      GO TO 300
  240 CAY12(II-IS.JJ-NUD) =-TEMP
  300 CONTINUE
C
C  NOW CALCULATE INDEPENDENT TERM G = K12 * (KNOWN DISPLACEMENTS)
C
      CALL MXMULT(CAY12.FD. G.I1.I4.1)
      RETURN
      END




      SUBROUTINE TMXMUL(A.B.C.N.M.K)
C
C  THIS SUBROUTINE MULTIPLIES MATRIX B BY THE TRANSPOSE OF MATRIX A
C  THE PRODUCT IS ADDED TO C
C
C    A IS (N X M)
C  B IS (N X K)
C  C IS (M X K)
C
      DIMENSION  A(N.M) . B(N.K) . C(M.K)
      DO 1  I=1.M
      DO 1  L=1.K
C     C(I.L) = 0.
      DO 1  J=1.N
      C(I.L) = C(I.L) + A(J.I) * B(J.L)
    1 CONTINUE
      RETURN
      END                                                    148 CARDS



      SUBROUTINE MXMULT(A.B.C.M.N.K)
C
C  THIS SUBROUTINE MULTIPLIES MATRIX A BY MATRIX B  AND STORES THE
C  PRODUCT IN C.   (C CANNOT BE THE SAME AS A OR B.)
C
C  A IS (M X N)
C  B IS (N X K)
C  C IS (M X K)
C
      DIMENSION  A(M.N) . B(N.K) . C(M.K)
C
      DO 1  I=1.M
      DO 1  L=1.K
      C(I.L) = 0.
      DO 1  J=1.N
      C(I.L) = C(I.L) + A(I.J) * B(J.L)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MXCON(A,B,X,M,N)
C
C  THIS SUBROUTINE MULTIPLIES MATRIX A (MXN) BY CONSTANT X, RESULT IN B
C  A MAY BE SAME AS B.
C  THIS SUBROUTINE MULTIPLIES MATRIX A (MXN) BY CONSTANT X, RESULT IN B
C  A MAY BE SAME AS B.
      DIMENSION  A(M,N) , B(M,N)
      DO 1 I=1,M
      DO 1 J=1,N
      B(I,J)= X*A(I,J)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MXSUB(A,B,C,M,N)
C
C  THIS SUBROUTINE SUBTRACTS MATRIX B FROM MATRIX A,STORES RESULT IN C
C
C  A, B, AND C ARE (M X N)        (C CAN BE THE SAME AS A OR B)
C
      DIMENSION  A(M,N) , B(M,N) ,C(M,N)
C
      DO 1  I=1,M
      DO 1  J=1,N
      C(I,J) = A(I,J) - B(I,J)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INVERT(B,K,K2,XM'N,FLAG)
C     THIS SUBROUTINE SETS UP A UNIT MATRIX ADJACENT TO B(K,K)
C     ELEMENTARY ROW OPERATIONS ARE THEN PERFORMED ON THE NEW K X 2K MATRIX
C     TO REDUCE B(K,K) TO A UNIT MATRIX. THIS WILL PLACE THE INVERSE OF
C     THE MATRIX B(K,K) IN THE RIGHT HALF OF B(K,2K)
C     ON EXIT, THE INVERSE OF B REPLACES B
C      B9+S N ARRAY OF 2*K**2 LOCATIONS CONTAINING THE MATRIX
C     K IS THE DIMENSION OF THE SQUARE MATRIX B
C     K2 IS 2*K
C     XMIN IS THE SMALLEST ALLOWABLE MAGNITUDE OF THE PIVOT
C     FLAG WILL BE RETURNED AS 0. IF THE INVERSION WENT OFF OK
C     FLAG WILL BE RETURNED AS 1.. IF A PIVOT ELEMENT WAS TOO SMALL
C     FLAG SHOULD BE TESTED AFTER EACH CALL TO THIS ROUTINE
C
      DIMENSION B(K,K2)
C
      FLAG = 0.
C
C     SET UP UNIT MATRIX
C
      IF(K.GT.1)  GO TO 20
      IF(ABS(B(1,1)).LT.XMIN)  GO TO 10
      B(1,1) = 1./B(1,1)
      RETURN
   20 CONTINUE
      DO 1  I=1,K
      DO 1  J=1,K
      B(I,K+J) = 0.
      IF(I.EQ.J)  B(I,K+J) = 1.
    1 CONTINUE
C
C     FIND LEADING ELEMENT WITH GREATEST MAGNITUDE
C
      DO 6  J=1,K
      M = J
      N = J+1
      IF(J.EQ.K) GO TO 21
      DO 2  L=N,K
      IF (ABS(B(M,J)).LT.ABS(B(L,J)))  M=L
    2 CONTINUE
   21 CONTINUE
      IF (ABS(B(M,J)).LT.XMIN)  GO TO 10
      IF(J.EQ.K) GO TO 31
C
C     INTERCHANGE JTH AND MTH ROWS
C
      DO 3  L=J,K2
      D = B(J,L)
      B(J,L) = B(M,L)
      B(M,L) = D
    3 CONTINUE
   31 CONTINUE
C
C     ZERO OUT PIVOTAL JTH COLUMN, SKIPPING PIVOTAL JTH ELEMENT
C
C     DIVIDE JTH ROW BY PIVOT
C
      DO 4  M=N,K2
      B(J,M) = B(J,M) / B(J,J)
    4 CONTINUE
      DO 6  M=1,K
```

```
C
C   M DETERMINES ROW BEING MODIFIED, ONE WHOLE ROW AT A TIME
C
      IF ( M.EQ.J )   GO TO 6
      DO 5  L=N,K2
C
C   L DETERMINES ELEMENT IN THE MTH ROW
C
      B(M,L) = B(M,L) - B(M,J) * B(J,L)
    5 CONTINUE
    6 CONTINUE
C
C   INVERSE OF B IS NOW IN RIGHT HALF OF B(K,K2)
C   NOW MOVE B INVERSE TO WHERE B WAS
      DO 7  I=1,K
      DO 7  J=1,K
      B(I,J) = B(I,J+K)
    7 CONTINUE
      RETURN
   10 FLAG = 10.
      RETURN
      END
```

DECK SET UP FOR TEN RUNS OF FUNDAMENTAL CASES ONE THROUGH THREE AND SIX

```
RUN     1
.5      30.      1.14E+7     .2     .5714285          1
RUN     2
.5      60.      2.28E+7     .2     .5714285          2
RUN     3
.5      90.      3.42E+7     .2     .5714285          3
RUN     4
.5      120.     4.56E+7     .2     .5714285          4
RUN     5
.5      160.     6.08E+7     .2     .5714285          5
RUN     6
.6      30.      1.14E+7     .2     .5714285          6
RUN     7
.6      60.      2.28E+7     .2     .5714285          7
RUN     8
.6      90.      3.42E+7     .2     .5714285          8
RUN     9
.6      120.     4.56E+7     .2     .5714285          9
RUN     10
.6      160.     6.08E+7     .2     .5714285          10
                                                (BLANK)
                                                (BLANK)
```

DECK SET UP FOR TEN RUNS OF FUNDAMENTAL CASES FIVE AND FOUR

```
  RUN      1                                                               1
   4.75E+6        1.4074E+5        .5
  RUN      2                                                               2
   9.50E+6        1.4074E+5        .5
  RUN      3                                                               3
  14.25E+6        1.4074E+5        .5
  RUN      4                                                               4
  19.00E+6        1.4074E+5        .5
  RUN      5                                                               5
25.33333E+6       1.4074E+5        .5
  RUN      6                                                               6
   4.75E+6        1.4074E+5        .6
  RUN      7                                                               7
   9.50E+6        1.4074E+5        .6
  RUN      8                                                               8
  14.25E+6        1.4074E+5        .6
  RUN      9                                                               9
  19.00E+6        1.4074E+5        .6
  RUN      10                                                              10
25.33333E+6       1.4074E+5        .6                                (BLANK)
                                                                     (BLANK)
```

DECK SET UP FOR RUNNING PHASE TWO AND PHASE THREE PROGRAMS. THIS DECK ASSUMES THE SAME TEN RUNS AS THE SAMPLE DATA FOR THE FUNDAMENTAL CASES.

.5	30.	1.14E+7	.2	.5714285	1
.5	60.	2.28E+7	.2	.5714285	2
.5	90.	3.42E+7	.2	.5714285	3
.5	120.	4.56E+7	.2	.5714285	4
.5	16[illegible].	6.[illegible]8E+7	.2	.5714285	5
.6	3[illegible].	1.14E+7	.2	.5714285	6
.6	5[illegible].	2.28E+7	.2	.5714285	7
.6	9[illegible].	3.42E+7	.2	.5714285	8
.6	120.	4.56E+7	.2	.5714285	9
.6	160.	6.[illegible]8E+7	.2	.5714285	10

20 CARDS

PHASES TWO AND THREE

```
      PROGRAM FENG
      COMMON /1/
     1      S(316,10,6),X(53),Y(53), NDIR(79,4),
     2            M,N,CENT(79,2), VF,E(2),G(2),XNU(2)   ,
     3  SMAC(6,13), SMIC(6,79),EN(79) ,GAMMA ,NS
      DATA  (PI=3.14159)
C
C  THIS PROGRAM READS STRESSES FROM TAPE 20 ASSUMED WRITTEN AS FOLLOWS
C
C      SIX FUNDAMENTAL CASES, EACH OF WHICH WAS RUN CONSECUTIVELY
C      NS TIMES.  EACH RUN PRODUCING STRESSES FOR
C      79 TRIANGLES.
C
C      NS IS THE FIRST RECORD ON THE TAPE (NS.LE.10)
C      FOLLOWING ARE NS RECORDS OF 316 STRESSES FOR EACH OF THE
C      FUNDAMENTAL CASES ONE THRU FOUR.
C      FOLLOWING ARE NS RECORDS OF 158 STRESSES FOR EACH OF THE
C      FUNDAMENTAL CASES FIVE THROUGH SIX
      REWIND 20
C
      DO 10 I=2,6,2
      DO 10 J=1,13
   10 SMAC(I,J) = 0.
      READ (20) NS
      PIO24=PI/24.
      DO 15 I=1,13
      ALP = (I-1)*PIO24
      SMAC(3,I) = COS(ALP)**2
      SMAC(1,I) = SIN(ALP)**2
   15 SMAC(5,I) = SIN(2.*ALP)/2.
      READ 1020,IJUMP
C
C   IJUMP = 1, FIRST TEN CASES
C   IJUMP = 2, SECOND TEN CASES
C
      GO TO (18,16),IJUMP
   16 REWIND 20
      CALL TAPESKIP(20,6,J)
   18 CONTINUE
      DO 30 K=1,4
      DO 20 J=1,NS
   20 READ (20) (S(I,J,K),I=1,316)
C
C   CROSS OVER THE EOF GAP
C
      READ (20) EOF
   30 CONTINUE
      DO 50 K=5,6
      DO 40 J=1,NS
   40 READ (20) (S(I,J,K),I=1,158)
      READ (20) EOF
   50 CONTINUE
C
C  NOW HAVE ALL RAW STRESS DATA FOR NS RUNS OF 6 FUNDAMENTAL CASES
      KASE = 0
```

```
C
      CALL TOPOL
      LINE = 4
      PRINT 1004,(J,J=1,3),(J,J=1,3)
      DO 130 I=1,57,2
      IF(LINE.LT.54) GO TO 120
      LINE = 4
      PRINT 1004,(J,J=1,3),(J,J=1,3)
  120 LINE = LINE + 2
      II=I+1
  130 PRINT 1005,I, (NDIR(I,J),J=1,4),II,(NDIR(II,J),J=1,4)
      I=79
      PRINT 1015,I,(NDIR(I,J),J=1,4)
C
      VF1 = 0.
  150 READ 1000, VF, GAMMA, E(1),XNU(1),BETA
      KASE = KASE+1
      IF(VF.EQ.0.)  GO TO 950
      E(2) = E(1)/GAMMA
      XNU(2) = XNU(1)/BETA
      G(1) = E(1)/(2.*(1.+XNU(1)))
      G(2) = E(2)/(2.*(1.+XNU(2)))
      PRINT 1001 ,VF,E(1),E(2),G(1),G(2),XNU(1),XNU(2)
      IF(ABS(VF-VF1).LT.1.E-6)  GO TO 350
      VF1 = VF
      CALL GEOM
C   PRINT OUT NODE INFORMATION
C
      I = 1
  190 CONTINUE
      LINE = 4
      PRINT 1007
  200 CONTINUE
      IF(I.NE.52) GO TO 210
      IP1 = I+1
      PRINT 1018, I, X(I), Y(I), IP1, X(IP1), Y(IP1)
      GO TO 220
  210 IP2 = I+2
      PRINT 1018,((J,X(J),Y(J)),J=I,IP2)
  215 I=I+3
      LINE = LINE + 2
      IF(LINE.GT.56) GO TO 190
      GO TO 200
  220 CONTINUE
C
  250 CONTINUE
      PRINT 1008
      DO 300 I=1,77,2
      I1 = I+1
      PRINT 1009,I,CENT(I,1),CENT(I,2),I1,CENT(I1,1),CENT(I1,2)
  300 CONTINUE
      I = 79
      PRINT 1009,I,CENT(I,1),CENT(I,2)
  350 CALL FIXSTR(KASE)
      CALL PHASE2(KASE)
      GO TO 150
  950 REWIND 20
      CALL PHASE3
      STOP
```

```
1000 FORMAT(5E10.3)
1001 FORMAT(1H1,15X,2HVF,13X,4HE(1),13X,4HE(2),13X,4HG(1),13X,4HG(2),
    1   12X,5HNU(1),12X,5HNU(2)/1X,7(4X,E13.6)     ////)
1004 FORMAT(1H1, 8HTRIANGLE,3(2X,5HNODE ,I1),4X,8HMATERIAL,10X,
    1              8HTRIANGLE,3(2X,5HNODE ,I1),4X,8HMATERIAL/)
1005 FORMAT(1X,2(5X,I3,5X,I3,5X,I3,5X,I3 ,9X,I3,10X)/)
1007 FORMAT(1H1,3(4HNODE,12X,1HX,12X,1HY,8X)/)
1008 FORMAT(1H1,2(8HTRIANGLE,5X,10HCENTROID X,5X,10HCENTROID Y)/)
1009 FORMAT(1H ,2(5X,I3,5X,E10.3,5X,E10.3))
1015 FORMAT(1X,4(5X,I3),9X,I3/)

1018 FORMAT(1X,3(1X,I3,3X,F10.5,3X,F10.5,8X)/)
1020 FORMAT(I1)
     END
```

```
      SUBROUTINE  MXSOL(B,K,K2,XMIN,FLAG)
C  THIS SUBROUTINE SOLVES  (K X K) SYSTEM OF EQUATIONS AND PLACES
C  THE RESULT IN THE (K+1ST) COLUMN OF B.  B IS A (K X K2) MATRIX,
C  WHERE K2 = K+1.
C  XMIN IS THE SMALLEST ALLOWABLE MAGNITUDE OF THE PIVOT
C  FLAG WILL BE RETURNED AS 0. IF THE INVERSION WENT OFF OK
C  FLAG WILL BE RETURNED AS 10. IF A PIVOT ELEMENT WAS TOO SMALL
C  FLAG SHOULD BE TESTED AFTER EACH CALL TO THIS ROUTINE
C
      DIMENSION B(K,K2)
C
      FLAG = 0.
C
C
C  FIND LEADING ELEMENT WITH GREATEST MAGNITUDE
C
      DO 6  J=1,K
      M = J
      N = J+1
      IF(N.GT.K)  GO TO 21
      DO 2  L=N,K
      IF (ABS(B(M,J)).LT.ABS(B(L,J)))  M=L
    2 CONTINUE
   21 CONTINUE
      IF (ABS(B(M,J)).LT.XMIN)  GO TO 10
C
C  INTERCHANGE JTH AND MTH ROWS
C
      DO 3  L=J,K2
      D = B(J,L)
      B(J,L) = B(M,L)
      B(M,L) = D
    3 CONTINUE
C
C  ZERO OUT PIVOTAL JTH COLUMN,  SKIPPING PIVOTAL JTH ELEMENT
C
C  DIVIDE JTH ROW BY PIVOT
C
      DO  4  M=N,K2
      B(J,M) = B(J,M) / B(J,J)
    4 CONTINUE
      DO 6  M=1,K
C
C  M DETERMINES ROW BEING MODIFIED, ONE WHOLE ROW AT A TIME
C
      IF ( M.EQ.J )   GO TO 6
      DO 5  L=N,K2
C
C  L DETERMINES ELEMENT IN THE MTH ROW
C
      B(M,L) = B(M,L) - B(M,J) * B(J,L)
    5 CONTINUE
    6 CONTINUE
C
      RETURN
   10 FLAG = 10.
      RETURN
      END
```

```
      SUBROUTINE TOPOL
C
C  THIS SUBROUTINE SETS UP THE TOPOLOGICAL RELATIONSHIPS FOR PROGRAM
C  FENG.  THESE REMAIN INVARIANT WHEN THE GEOMETRY CHANGES WITH VF.
C
C
C
      COMMON /1/
     1      S(316,10,6),X(53),Y(53), NDIR(79,4),
     2              M,N,CENT(79,2), VF,E(2),G(2),XNU(2)   ,
     3  SMAC(6,13), SMIC(6,79),EN(79) ,GAMMA ,NS
      DIMENSION  JNDIR(79) , KNDIR(79) , LNDIR(79)
      DATA (JNDIR =
     1   1,1,1,2,2,3,3,3,4,4,4,5,6,7,7,8,8,8,9,10,10,11,11,12,13,13,
     2   14,15,15,16,16,17,17,18,18,19,20,23,24,24,25,25,25,26,28,29,
     3   29,30,30,35,21,21,21,21,22,22,23,23,27,27,27,28,28,36,46,37,
     4   47,38,48,39,39,40,41,41 41,42,42,43,43)
      DATA (KNDIR =
     5   2,3,4,6,7,7,8,9,9,10,11,11,13,6,14,14,15,16,16,16,17,10,
     6   18,18,20,21,21,21,22,15,23,23,24,17,25,25,35,27,27,28,28,29,
     7   30,30,32,32,33,33,34,36,35,37,38,39,39,40,40,41,41,42,43,43,44,
     8   45,37,46,38,47,39,48,49,49,40,50,51,51,52,52,53)
      DATA (LNDIR =
     9   3,4,5,7,3,8,9,4,10,11,5,12,14,14,8,15,16,9,10,17,18,18,12,
     1   19,21,14,15,22,23,23,17,24,25,25,19,26,21,24,28,25,29,30,26,
     2   31,29,33,30,34,31,37,37,38,39,22,40,23,41,27,42,43,28,44,32,
     3   46,36,47,37,48,38,49,40,50,50,51,42,52,43,53,44 )
      DO 10 I=1,36
   10 NDIR(I,4) = 1
      DO 20 I=37,79
      NDIR(I,4) = 2
   20 CONTINUE
      DO 100  I = 1, 79
      NDIR(I,1) = JNDIR(I)
      NDIR(I,2) = KNDIR(I)
      NDIR(I,3) = LNDIR(I)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GEOM
C   THIS SUBROUTINE IS CALLED TO GENERATE NEW COORDINATES AND CENTROIDS
C   WHENEVER A NEW VALUE OF VF IS CONSIDERED IN PROGRAM FENG.
C
      COMMON /1/
     1      S(316,10,6),X(53),Y(53), NDIR(79,4),
     2              M,N,CENT(79,2), VF,E(2),G(2),XNU(2)   ,
     3  SMAC(6,13), SMIC(6,79),EN(79) ,GAMMA ,NS
      COMMON /2/  X1(6),Y1(6)
      DIMENSION  NOTRI(6,2)
      DATA  (NOTRI = 25, 28, 29, 32, 33, 36,
     1             37, 54, 56, 38, 40, 43)
      DATA  (RAD = 57.29578)
      S3 = SQRT(3.)
      S302 = S3/2.
      PI = 3.1415927
      R = SQRT(2.*S3*VF/PI)
      X(1) = S302
      Y(1) = .5
      DO 210 I=1,4
      X(I+1) = S302 - R/4.*COS(PI*(I-1)/6.)
      Y(I+1) = .5 -  R/4. *SIN(PI*(I-1)/6.)
  210 CONTINUE
C
      DO 220  I=1,7
      X(I+5) = S302 - R/2.* COS(PI*(I-1)/12.)
      Y(I+5) = .5   - R/2.* SIN(PI*(I-1)/12.)
C
      X(I+12) =  S302 - 3.*R/4.*COS(PI*(I-1)/12.)
      Y(I+12) =  .5  -  3.*R/4.*SIN(PI*(I-1)/12.)
C
      X(I+19) =  S302 - R * COS(PI*(I-1)/12.)
      Y(I+19) =  .5  -  R * SIN(PI*(I-1)/12.)
  220 CONTINUE
      X(34) = S302
      Y(34) = - .5
      X(31) = S302
      Y(31) = (Y(26)+Y(34)) / 2.
      X(45) = -.5* TAN(PI/6.)
      Y(45) = .5
      DX = (1.-R)/(2.*COS(PI/6.))
      X(36) = X(45) + DX
      Y(36) = .5
      X(35) = (X(20)+X(36))/2.
      Y(35) = .5
C
      DO 230 I = 1,8
      X(I+45) = (4 -I)* X(45) /4.
      Y(I+45) = (4 -I)* Y(45) /4.
  230 CONTINUE
C
      DELX = X(46) -X(45)
      DELY = Y(46)-Y(45)
      DO 240 I = 1,8
      X(I+36) = X(I+35) +DELX
      Y(I+36) = Y(I+35) +DELY
  240 CONTINUE
      X(32) = X(44) +(X(34)-X(44))/3.
      Y(32) = -.5
      X(33) = 2.*X(32) - X(44)
      Y(33) = -.5
      X(27) = X(23) + (X(32)-X(23))/3.
```

```
      Y(27) = Y(23) + (Y(32)-Y(23))/3.
      X(28) = 2.*X(27)-X(23)
      Y(28) = 2.*Y(27)-Y(23)
      X(29) = X(28) +(X(31)-X(28))/3.
      Y(29) = Y(28) +(Y(31)-Y(28))/3.
      X(30) = 2.*X(29)-X(28)
      Y(30) = 2.*Y(29)-Y(28)
C
C
      DO 200 I=1,79
      NDIR1 = NDIR(I,1)
      NDIR2 = NDIR(I,2)
      NDIR3 = NDIR(I,3)
      CENT(I,1) = (X(NDIR1)+X(NDIR2)+X(NDIR3))/3.
  200 CENT(I,2) = (Y(NDIR1)+Y(NDIR2)+Y(NDIR3))/3.
      DO 300 I=1,6
      X1(I) = (X(I+19)+X(I+20))/2.
  300 Y1(I) = (Y(I+19)+Y(I+20))/2.
      RETURN
      END
```

```
      SUBROUTINE PHASE2(J)
C
C   THIS SUBROUTINE CALCULATES AND PRINTS THE PARAMETERS REQUIRED
C   IN PHASE II.  ONE CALL TO THIS SUBROUTINE PRODUCES 12 BLOCKS OF 79
C   SETS OF DATA.  ONE FOR EVERY TRIANGLE FOR EACH OF 12 VALUES OF
C   ALPHA.
C
C    K   COUNTS ALPHA
C    I   COUNTS TRIANGLES
C    M   COUNTS FUNDAMENTAL CASES
C    L   COUNTS POSITIONS WITHIN MICROSTRESS VECTOR
C    J   RUN NUMBER
C
      COMMON /1/
     1      S(316,10,6),X(53),Y(53), NDIR(79,4),
     2            M,N,CENT(79,2), VF,E(2),G(2),XNU(2)   ,
     3  SMAC(6,13), SMIC(6,79),EN(79) ,GAMMA ,NS
      DATA  (SIGFT2=7000.) , (SIGFC2=17000.)
      IF(ABS(GAMMA-  1.).LE.1.)  GO TO 100
      IF(ABS(GAMMA- 30.).LE.1.)  GO TO 120
      IF(ABS(GAMMA- 60.).LE.1.)  GO TO 140
      IF(ABS(GAMMA- 90.).LE.1.)  GO TO 160
      IF(ABS(GAMMA-120.).LE.1.)  GO TO 180
      IF(ABS(GAMMA-160.).LE.1.)  GO TO 200
  100 SIGFT1 = 7000.
      SIGFC1 = 17000.
      GO TO 250
  120 SIGFT1 = 160000.
      SIGFC1 = 160000.
      GO TO 250
  140 SIGFT1 = 200000.
      SIGFC1 = 200000.
      GO TO 250
  160 SIGFT1 = 250000.
      SIGFC1 = 250000.
      GO TO 250
  180 SIGFT1 = 300000.
      CIGFC1 = 300000.
      GO TO 250
  200 SIGFT1 = 350000.
      SIGCT1 = 350000.
  250 CONTINUE
      DO 500 K=1,13
      ALP = 7.5*(K-1)
      DO 350 I=1,79
      I4 = 4*(I-1)
      I2=2*(I-1)
      DO 260 L=1,4
      SMIC(L,I) = 0.
      DO 260 M=1,4
  260 SMIC(L,I)= SMIC(L,I)+S(I4+L,J,M)*SMAC(M,K)
      DO 270 L=1,2
      SMIC(L+4,I) = 0.
      DO 270 M=5,6
  270 SMIC(L+4,I) = SMIC(L+4,I)+S(I2+L,J,M)*SMAC(M,K)
      IF(I.LE.36) GO TO 300
      SXYZ = SMIC(1,I)+SMIC(2,I)+SMIC(3,I)
      IF(SXYZ.GE.0.) GO TO 280
      SIGF = SIGFC2
      GO TO 290
  280 SIGF = SIGFT2
  290 GO TO 320
```

```
300 SXYZ = SMIC(1,I)+SMIC(2,I)+SMIC(3,I)
    IF(SXYZ.GE.0.) GO TO 310
    SIGF = SIGFC1
    GO TO 320
310 SIGF = SIGFT1
320 EN(I)=SIGF*SQRT(2./((SMIC(1,I)-SMIC(2,I))**2
   1      +(SMIC(2,I)-SMIC(3,I))**2+(SMIC(3,I)-SMIC(1,I))**2
   2      +6.*(SMIC(4,I)**2+SMIC(5,I)**2+SMIC(6,I)**2)))
350 CONTINUE
    PRINT 1002,ALP
    PRINT 1000
    DO 400 N=1,79
    IF(N.NE.54) GO TO 400
    PRINT 1003
    PRINT 1000
400 PRINT 1001, N, (SMIC(JJ,N),JJ=1,6),EN(N)
    CALL INTERF(K,J)
500 CONTINUE
1000 FORMAT(9H TRIANGLE,8X,7HSIGMA X,8X,7HSIGMA Y,8X,7HSIGMA Z,
   1      9X,6HTAU XY,9X,6HTAU XZ,9X,6HTAU YZ,14X,1HN/)
1001 FORMAT(6X,I3,7(5XE10.3))
1002 FORMAT(8H1THETA =,F10.3/)
1003 FORMAT(1H1)
    RETURN
    END
```

```
      SUBROUTINE PHASE3
C
C  THIS SUBROUTINE READS AND PRINTS ALL PHASE III DATA FROM TAPE 96
C
      COMMON /1/
     1      S(316,10,6),X(53),Y(53), NDIR(79,4),
     2             M,N,CENT(79,2), VF,E(2),G(2),XNU(2)   ,
     3  SMAC(6,13), SMIC(6,79),EN(79) ,GAMMA ,NS
      DIMENSION  ENS(4,5,13)
      DIMENSION SM(6,4)
      REWIND 96
      IVF = 0
      VF1 = 0.
      GAMMA1 = 1000.
      PRINT 1000
      NPAGE =(NS*13)/2
C
C    NS ASSUMED EVEN
C
      DO 300 IP=1,NPAGE
      PRINT 1001
      DO 300 J=1,2
      READ (96) VF,EI,EII,GI,GII,XNUI,XNUII,N,ENI,(SM(K,1),K=1,6),
     1    M,ENII,(SM(K,2),K=1,6), X1,Y1,EN1,SNAV,
     2   X2,Y2,EN2,TAUTAV,I
      GAMMA = EI/EII
      IF(I.NE.1) GO TO 100
      IF(GAMMA.LE.GAMMA1+.5) GO TO 60
      IGAM = IGAM+1
      GO TO 100
   60 IVF = IVF+1
      IGAM = 2
      GAMMA1 = GAMMA
  100 ENST = ENI
      IF(ENST.GT.ENII)  ENST = ENII
      IF(EN1.LE.-1.E+8) GO TO 110
      IF(ENST.GT.EN1) ENST = EN1
  110 IF(ENST.GT.EN2) ENST = EN2
      ENS(IVF,IGAM,I) = ENST
      ALP = 7.5*(I-1)
      PRINT 1002
      PRINT 1003, VF,EI,EII,GI,GII,XNUI,XNUII,ALP
      PRINT 1004
      PRINT 1005,(SMAC(K,I),K=1,6)
      PRINT 1006
      PRINT 1007, N,ENI,(SM(K,1),K=1,6)
      PRINT 1008
      PRINT 1007, M,ENII,(SM(K,2),K=1,6)
      PRINT 1009
      IF(EN1.LT.-1.E+8) GO TO 180
  150 CONTINUE
      PRINT 1010
      PRINT 1017, X1,Y1,EN1,SNAV
  180 CONTINUE
      PRINT 1011
      PRINT 1017, X2,Y2,EN2,TAUTAV
      PRINT 1013
  300 CONTINUE
      DO 400 I=1,2
      DO 400 J=2,IGAM
      DO 400 K=1,13
C
```

```
C    INDEX 1 FOR GAMMA IS SAVED FOR HOMOGENEOUS CASE, RUN SEPARATELY
C
      PUNCH 2000,I,J,K,ENS(I,J,K)
  400 CONTINUE
 1000 FORMAT(1H1,50X,9HPHASE III)
 1001 FORMAT(1H1)
 1002 FORMAT(13X,2HVF,12X,3HE I,11X,4HE II,12X,3HG I,11X,4HG II,
     1   11X,4HNU I,10X,5HNU II,10X,5HTHETA)
 1003 FORMAT(8(5X,E10.3)/)
 1004 FORMAT(1X,13X,2HNX,13X,2HNY,13X,2HNZ,12X,3HTXY,12X,3HTXZ,
     1     12X,3HTXZ)
 1005 FORMAT(1X,6(5X,E10.3)/)
 1006 FORMAT(1X,8HTRIANGLE,4X,12HSMALLEST NI ,8X,7HSIGMA X,8X,7HSIGMA Y,
     1      8X,7HSIGMA Z,9X,6HTAU XY,9X,6HTAU XZ,9X,6HTAU YZ)
 1007 FORMAT(1X,6X,I2,6X,E10.3,6(5X,E10.3)/)
 1008 FORMAT(1X,8HTRIANGLE,4X,12HSMALLEST NII,8X,7HSIGMA X,8X,7HSIGMA Y,
     1      8X,7HSIGMA Z,9X,6HTAU XY,9X,6HTAU XZ,9X,6HTAU YZ)
 1009 FORMAT(50X,9HINTERFACE)
 1010 FORMAT(8X,1HX,11X,1HY,4X,12H SMALLEST N1,3X,15HAVERAGE SIGMA N)
 1011 FORMAT(8X,1HX,11X,1HY,4X,12H SMALLEST N2,3X,15H  AVERAGE TAU T)
 1013 FORMAT(//)
 1017 FORMAT(1X,F8.3,4X,F8.3,2E16.3/)
 1020 FORMAT(8X,1HX,11X,1HY,1X,15HAVERAGE SIGMA N)
 2000 FORMAT(3I3,E15.8)
      STOP
      END
```

```
      SUBROUTINE FIXSTR(KASE)
C
C  THIS SUBROUTINE MODIFIES THE STRESSES S(I,J,K), I=1,316, K=1,4 ,
C       RUN= KASE , PRODUCED BY THE NS RUNS OF FUNDAMENTAL CASES ONE
C           THRU FOUR.
C
      COMMON /1/
     1      S(316,10,6),X(53),Y(53), NDIR(79,4),
     2              M,N,CENT(79,2), VF,E(2),G(2),XNU(2)   ,
     3  SMAC(6,13), SMIC(6,79),EN(79) ,GAMMA ,NS
      DIMENSION    A(3,4,3),ABC(3,3),AREA(79), Z(3,4), ARTOP(11),AREND(6)
      DIMENSION    PTEND(7), PTTOP(13),TRITOP(11),TRIEND(6), TEMP(316,4)
      TYPE INTEGER PTEND,TRIEND,PTTOP,TRITOP
      DATA        (TRITOP=1,4,13,25,37,50,64,79,63,46,48)
      DATA        (TRIEND=3,12,24,36,44,49)
      DATA        (PTEND=1,5,12,19,26,31,34)
      DATA        (PTTOP=1,2,6,13,20,35,36,45,53,44,32,33,34)
      DO  50 I=1,7
      J1 = PTTOP(I)
      J2 = PTTOP(I+1)
   50 ARTOP(I)  = X(J1)-X(J2)
      DO 100 J=9,12
      J1=PTTOP(J)
      J2=PTTOP(J+1)
  100 ARTOP(J-1) = X(J2)-X(J1)
      DO 110 I= 1,6
      J1 = PTEND(I)
      J2 = PTEND(I+1)
  110 AREND(I)=Y(J1)-Y(J2)
C
      DO 120 I= 1,79
      J1 = NDIR(I,1)
      J2 = NDIR(I,2)
      J3 = NDIR(I,3)
  120 AREA(I) =  X(J1)*(Y(J2)-Y(J3))+X(J2)*(Y(J3)-Y(J1))
     1                                   +X(J3)*(Y(J1)-Y(J2))
      DO 130 I=1,3
      DO 130 J=1,4
  130 Z(I,J) = 0.
      DO 150 I= 1,6
      I1 = TRIEND(I)
      I2 = (I1-1)*4 +1
      DO 150 J= 1,3
  150 Z(1,J)= Z(1,J)+AREND(I )*S(I2,KASE,J)
C
      DO 160 I=1,11
      I1 = TRITOP(I)
      I2 = (I1-1)*4 +2
      DO 160 J=1,3
  160 Z(2,J) = Z(2,J)+ARTOP(I )*S(I2,KASE,J)
C
      DO 180 I =1,79
      I2 = (I-1)*4+3
      DO 180 J =1,3
  180 Z(3,J) = Z(3,J)+AREA(I)*S(I2,KASE,J)
C
C  NOTE THAT AREA ALREADY CONTAINS A FACTOR OF TWO FROM ITS DERIVATION
C
      DO 300 I=1,3
      DO 200 J=1,3
      DO 200 K=1,4
  200 A(J,K,I) = Z(J,K)
```

```
      GO TO(210,220,230),I
  210 A(1,4,1) = 1.
      GO TO 240
  220 A(2,4,2) = SQRT(3.)
      GO TO 240
  230 A(3,4,3) = SQRT(3.)
  240 CALL MXSOL(A(1,1,I),3,4,1.E-7,FLAG)
      IF(FLAG.NE.0.) GO TO 600
      DO 250 J=1,3
  250 ABC(J,I) = A(J,4,I)
C
C ABC(I,1) CONTAINS A(I) , I=1,2,3
C ABC(I,2) CONTAINS B(I) , I=1,2,3
C ABC(I,3) CONTAINS C(I) , I=1,2,3
C
      DO 280 J=1,316
  280 TEMP(J,I)= ABC(1,I)*S(J,KASE,1)+ABC(2,I)*S(J,KASE,2)
     1          + ABC(3,I)*S(J,KASE,3)
  300 CONTINUE
      DO 400 I=1,316
      DO 400 J=1,3
  400 S(I,KASE,J) = TEMP(I,J)
      RETURN
  600 PRINT 1000, KASE, I
      STOP
 1000 FORMAT(14H1BLEW UP,  RUN,I3,6X,16HFUNDAMENTAL CASE,I3)
      END
```

```
      SUBROUTINE INTERF(I,L)
C
C   I -- INDEX OF ALPHA,   I=1,13
C  L  -- RUN NUMBER  (DETERMINED BY VF,EI COMBINATION)
C
      COMMON /1/
     1     S(316,1,6),X(53),Y(53), NDIR(79,4),
     2              M,N,CENT(79,2), VF,E(2),G(2),XNU(2)   ,
     3  SMAC(6,13), SMIC(6,79),EN(79) ,GAMMA ,NS
      COMMON /2/  X1(6),Y1(6)
      DIMENSION  SNI(6),SNII(6), TAUTI(6),TAUTII(6), SNAV(6),TAUTAV(6)
      DIMENSION  EN1(6), EN2(6), K1(6), K2(6)
      DIMENSION  NOTRI(6,2)
C
C  NOTRI(M,N) --  NTH INTERFACE TRIANGLE, MATERIAL M
C
      DATA (SNF=10000.), (TAUTF = 10000.)
      DATA  (NOTRI = 25, 28, 29, 32, 33, 36,
     1                 37, 54, 56, 38, 40, 43)
      DATA  (PI=3.14159   )
C
      DO 100 K=1,6
      K1(K) = NOTRI(K,1)
      K2(K) = NOTRI(K,2)
      L1 = K1(K)
      L2 = K2(K)
      PHI = PI*(K-.5)/12.
      CPHI = COS(PHI)
      SPHI = SIN(PHI)
      S2PHI = SIN(2.*PHI)
      C2PHI = COS(2.*PHI)
      SNI(K)=CPHI**2*SMIC(1,L1)+SPHI**2*SMIC(2,L1)
     1     +2.*SMIC(4,L1)*SPHI*CPHI
      SNII(K)=CPHI**2*SMIC(1,L2)+SPHI**2*SMIC(2,L2)
     1     +2.*SMIC(4,L2)*SPHI*CPHI
      TAUTI(K)=SQRT(((SMIC(1,L1)-SMIC(2,L1))*S2PHI/2.-SMIC(4,L1)*C2PHI)
     1   **2+(CPHI*SMIC(5,L1)+SPHI*SMIC(6,L1))**2)
      TAUTII(K)=SQRT(((SMIC(1,L2)-SMIC(2,L2))*S2PHI/2.-SMIC(4,L2)*C2PHI)
     1   **2+(CPHI*SMIC(5,L2)+SPHI*SMIC(6,L2))**2)
      SNAV(K) = (SNI(K)+SNII(K))/2.
      TAUTAV(K) = (TAUTI(K)+TAUTII(K))/2.
      EN1(K) = SNF/  SNAV(K)
      EN2(K) = TAUTF/TAUTAV(K)
  100 CONTINUE
      PRINT 1000
      PRINT 1001
      DO 150 K=1,6
  150 PRINT 1002,K1(K),SNI(K),TAUTI(K),K2(K),SNII(K),TAUTII(K)
      PRINT 1003
      IF(SNI( 1))210,190,190
  190 PRINT 1004
      DO 200 K=1,6
  200 PRINT 1005,K,SNAV(K),TAUTAV(K),EN1(K),EN2(K)
      GO TO 230
  210 PRINT 1014
      DO 220 K=1,6
  220 PRINT 1005,K,SNAV(K),TAUTAV(K),EN2(K)
  230 CONTINUE
C
C   CALCULATE AND WRITE ON TAPE THE NECESSARY PHASE III VALUES
C
      N = 1
```

```
      DO 250 J=2,36
      IF(EN(J).GE.EN(N)) GO TO 250
      N=J
  250 CONTINUE
      M = 37
      DO 260 J=38,79
      IF(EN(J).GE.EN(M)) GO TO 260
      M=J
  260 CONTINUE
      L1 = 10
      ENWUN = 10.E+10
      DO 265 J=1,6
      IF(EN1(J).GT.ENWUN.OR.EN1(J).LT.0.) GO TO 265
      L1 = J
      ENWUN = EN1(J)
  265 CONTINUE
      IF(L1.LT.7) GO TO 270
      L1 = 1
      EN1(L1) = -10.E+8
  270 CONTINUE
      L2 = 1
      DO 280 J=2,6
      IF(EN2(J).GE.EN2(L2)) GO TO 280
      L2 = J
  280 CONTINUE
      WRITE (96) VF,E(1),E(2),G(1),G(2),XNU(1),XNU(2),
     1  N, EN(N),(SMIC(J,N),J=1,6), M, EN(M),(SMIC(J,M),J=1,6),
     2  X1(L1),Y1(L1),EN1(L1),SNAV(L1),
     3  XI(L2),Y1(L2),EN2(L2),TAUTAV(L2),I
 1000 FORMAT(///50X,9HINTERFACE)
 1001 FORMAT(27X,5HFIBER,54X,5HRESIN/1X,2(8HTRIANGLE,8X,7HSIGMA N,
     1    10X,5HTAU T,20X)/)
 1002 FORMAT(1X,2(5X,I3,5X,E10.3,5X,E10.3,20X) )
 1003 FORMAT(/50X,7HAVERAGE/)
 1004 FORMAT(1X,8HPOSITION,13X,7HSIGMA N,15X,5HTAU T,18X,2HN1,18X,2HN2/)
 1005 FORMAT(1X,7X,I1,4(10X,E10.3))
 1014 FORMAT(1X,8HPOSITION,13X,7HSIGMA N,15X,5HTAU T,18X,2HN2/)
      RETURN
      END
```

ENGINEERING CONSTANTS

```
100 DIMENSION E33(13<20),E11(13,20),G31(13,20),ANU31(13,20),
110 & ANU13(13,20),EP(13,20),GP(13,20),ANUP(13,20),ETA(13,20),
120 & CC(13,20),SS1(13,20),SC(13,20),SS(13,20),OC(13,20),OS(13,20)
130 DATA E33/13*6.110E6,13*1.176E7,13*1.739E7,13*2.303E7,13*3.055E7,
140 & 13*7.229E6,13*1.401E7,13*2.078E7,13*2.754E7,13*3.656E7,
150 & 13*8.366E6,13*1.629E7,13*2.419E7,13*3.208E7,13*4.260E7,
160 & 13*9.548E6,13*1.863E7,13*2.767E7,13*3.670E7,13*4.874E7/
170 & E11/13*1.869E6,13*1.945E6,13*1.971E6,13*1.985E6,13*1.995E6,
180 & 13*2.307E6,13*2.439E6,13*2.487E6,13*2.511E6,13*2.530E6,
190 & 13*3.[illegible]1E6,13*3.270E6,13*3.367E6,13*3.418E6,13*3.458E6,
200 & 13*4>334E6,13*4.975E6,13*5.237E6,13*5.380E6,13*5.492E6/
210 & G31/13*5.250E5,13*5.449E5,13*5.519E5,13*5.555E5,13*5.582E5,
220 & 13*6.749E5,13*7.103E5,13*7.230E5,13*7.295E5,13*7.345E5,
230 & 13*9.167E5,13*9.868E5,13*1.013E6,13*1.026E6,13*1.037E6,
240 & 13*1.392E6,13*1.571E6,13*1.641E6,13*1.679E>,13*1.709E6/
250 & ANU31/65*.293,65*>272,65*.256,65*.232/
300 DO 10 I=1,20
310 DO 20 J=1,13
320 BJ=J-1
330 THETA=BJ*3.141593/24.
340 CC(J,I)=COS(THETA)
350 SS1(J,I)=SIN(THETA)
360 SC(J,I)=(COS(THETA))**2
370 SS(J,I)=(SIN(THETA))**2
380 OC(J,I)=(COS(THETA))**4
390 OS(J,I)=(SIN(THETA))**4
400 ANU13(J,I)=ANU31(J,I)*E11(J,I)/E33(J,I)
500 EP(J,I)=1./(OC(J,I)/E33(J,I)+OS(J,I)/E11(J,I)+(1./G31(J,I)
510 & =ANU31(J,I)/E33(J,I)-ANU13(J,I)/E11(J,I))*SC(J,I)*SS(J,I))
600 GP(J,I)=1./(1./G31(J,I)+4.*SC(J,I)*SS(J,I)*((1.+ANU31(J,I))/
610 & E33(J,I)+(1.+ANU13(J,I))/E11(J,I)-1./G31(J,I)))
700 ANUP(J,I)=EP(J,I)*(ANU31(J,I)/E33(J,I)-SC(J,I)*SS(J,I)*
710 & ((1.+ANU31(J,I))/E33(J,I)+(1.+ANU13(J,I))/E11(J,I)-1./G31(J,I)))
800 ETA(J,I)=EP(J,I)*CC(J,I)*SS1(J,I)*(2.*SC(J,I)/E33(J,I)
810 & -2.*SS(J,I)/E11(J,I)+(SC(J,I)-SS(J,I))*(ANU31(J,I)/E33(J,I)
820 & /ANU13(J,I)/E11(J,I)-1./G31(J,I)))
900 20 CONTINUE
910 PRINT,(EP(J,I)/3.8E5,GP(J,I)/1.407E5,ANUP(J,I),ETA(J,I),J=1,13)
920 10 CONTINUE
1000 30 STOP;END
```

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Whittaker Corporation Research & Development Division 3540 Aero Court, San Diego, California	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

STRESS DISTRIBUTION AND MATERIAL CHARACTERISTICS OF COMPOSITE MATERIALS UNDER OBLIQUE LOADINGS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

FINAL REPORT

5. AUTHOR(S) (First name, middle initial, last name)

JUAN HAENER
MING-YUAN FENG

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
July 1970	181	14

8a. CONTRACT OR GRANT NO. DAAJ02-69-C-0029 b. PROJECT NO. Task 1F162204A17002 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) USAAVLABS Technical Report 70-15 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) WRD Final Report No. 3570

10. DISTRIBUTION STATEMENT

This document is subject to special export controls, and each transmittal to foreign governments or foreign nationals may be made only with prior approval of U. S. Army Aviation Materiel Laboratories, Fort Eustis, Virginia 23604.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U. S. Army Aviation Materiel Laboratories Fort Eustis, Virginia

13. ABSTRACT

The critical loadings at failure of a unidirectional fibrous composite under any oblique loading are the most important results obtained in this work. For this purpose the von Mises energy criterion was applied to the components of the composite. Debonding failures at the interfaces were also considered.

Other results of this research are the composite elastic engineering constants in any angular direction as a function of fiber density and component properties.

Diagrams are presented which exhibit the critical loading and the elastic coefficients of a series of composites as a function of the loading direction, component material constants, and geometry.

The fundamentals to this work are based on the micromechanical stress fields in the fibers and in the matrix.

DD FORM 1 NOV 65 1473

Unclassified
Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Oblique Loading of Composite Material						
Fundamental Loadings						
Elastic Constants						
Stress Distribution						
Fiber Orientation						

Unclassified

Security Classification